Passports/AAI Technical working Subgroup

Security and DURI Work Streams

Minutes and Actions 2021

Minute document for 2022:
https://docs.google.com/document/d/14bHDgsv1rWSHRgHCxgkD39z_FXlHJrE08ENQDMUmGwY/edit#

Overview

These are the minutes for the Passports/AAI Technical Working subgroup, a collaboration between the Data Security and DURI Work Streams.

This group was formed with the aim of creating a small technical task force to discuss outstanding technical issues regarding v1.0 and migrating to a new version of the GA4GH Passports Standard named v1.2. V1.1 was designated as an experimental release driven within the NIH, known as NIH RAS (details below).

The group aims to produce a summary document and specification to be shared more widely for further feedback and input from various groups across GA4GH including current and future implementers of the GA4GH Passports Standards.

The group consists of AAI and Passports leads, those who have been involved in the development of v1.0 from the start and those newer to the work but with an interest in the technical aspects of both AAI and Passport specifications.

Start-Up Guide to GA4GH Passports and AAI specifications

Passports

- Documentation for adopters
- Repository
- Use cases

AAI

- Specification
- Documentation for adopters
- Repository

Passports v1.2 work so far (output from this group)

- Summary current status of Passports v1.2 discussion

- Use cases from 2021

Table of Contents

Overview

Start-Up Guide to GA4GH Passports and AAI specifications

Document Links

Meeting Protocols

Meeting Details:

Meeting Minute Report

- 2021-12-16:
- 2021-12-02:
- 2021-11-18: “Multiple passports/visas approach in /token” vs the original proposed “compact custom format”.
- 2021-11-11: Further discussion on compact and proposal of multiple
- 2021-11-04: v1.2 plain text format
- 2021-10-28: Migration story, versions and v1.2 plain text format
- 2021-10-21: Token Endpoint, Architectural decision register and Version discussion
- 2021-10-07:
- 2021-09-23: Spec Update and Audience Discussion
- 2021-09-16: Spec Update and Audience Discussion
- 2021-09-09:
- 2021-09-02:
- 2021-08-26:
- 2021-08-12: Exploring proposed 4k passports
- 2021-08-12: Proposal of a smaller 4K passport
- 2021-08-03 : Pros and Cons for each of the two alternative options
- Minutes
- 2021-07-29: Token exchange mechanisms
- 2021-07-23: Second AAI/Passport hackathon concentrated group meeting
- 2021-06-29: First AAI/Passport hackathon meeting

Document Links

- Summary of discussions October 2021
    - Summary current status of Passports v1.2 discussion
- October Connect (2021) Passports Workshop Agenda and Minutes
- October Connect (2021) Workshop recordings here
- Use cases from 2021

Meeting Protocols

- Please note that by participating in meetings, attendees agree to adhere to the GA4GH Standards of Professional Conduct.
- Meetings may be recorded for note-taking purposes. Recordings will be deleted within three months of the meeting taking place.
- Dates should be specified in the ISO format yyyy-mm-dd

Meeting Details:

- To join the call please use Zoom
- We will use the Data Security Zoom account

Please use this link:
https://us02web.zoom.us/j/87148565813?pwd=SVVQU3c1SXdIclpQZGRTTDJ6RUhqdz09

GA4GH have added password protection:
Zoom Password = ga4gh
Meeting ID: 871 4856 5813

- Mailing list aai@ga4gh.org (To join this group please contact Alice Mann alice.mann@ga4gh.org)

Meeting Minute Report

2021-12-16:

Chair:
Attendees - Name (Affiliation): Kurt Rodarmer (NIH), Martin Kuba (ELIXIR-CZ), David Bernick, Mikael Linden

Apologies: Susan Fairley (GA4GH), Max Barkley (DNAstack), Andrew Patterson (UMCCR)

	Actions Arising	Assigned To	Deadline
1			
2			
3			
4			
5.			
6.			

	Agenda Item	Person/Time
1.		
2.		
3.		
4.		

Minutes

- KR: my vote is landing v1.2, I don't think it is substantially different from where we are today, apart from the size. All of the other things we are discussing conceptually, how the overall passport fits our models
- I think we can continue next year, i don't think avoiding that we've talked about for v1.2 does anything to further those ideas
- MK: disagree with this, problem we want to solve has nothing to do with passport.
- See the diagram from MK, we had some discussion in ELIXIR to explain what we are talking about. I asked why EGA does not have similar problems.
- DACs decisions into DbGAP provides visa into passport broker and gives it to user interface, then some WES and TES to data retrieval service.. NIH needs something

represented by green pentagon to get it from dbGAP into DRS or into some system that provides the data.

- EGA have no TES of their own which would need to access the data and also they don't need to pass around some permissions because all they do is they have access token which contains the userID and all they need is this userID so they are somehow passing the info locally need from their database of decisions to their DRS
- MK: there is communication between PEP and dbGAP block?
- MK: i think EGA is doing this, so why can NIH not do the same as EGA?
- TC: don't know nearly as much as this as Kurt but it strikes me as from a passport perspective trying to solve for when the permissions are coming from dbGAP or NIH and then TES submitting jobs to som other place where the data is held, governed by NIH but held in a different place.
- KR: skeptical about EGA approach and their needs. An answer given by someone who doesn't understand the security behind the system.all they are trying to do is preserve an identity and perform another identity based access without any concern for who validated the identity or how any of that works, maybe i am wrong.
- What NIH is a flow of authorisation, different to flow of identity. You do or may preserve the identity from the origin to the end of the flow. It is the authorisation that flows not the identity. Identity based systems have issues in federation as requires everyone working on user identity and user databases. Model you have, doesn't provide for permission scoping, what if i had as user access to the entire holdings of EGA and TES and TES only intended to access two files yet system based on identity as described (Oauth scopes in practice don't actually describe downscoping, no provision scope for it in GA4GH other than entire passport), can still ask for anything to ask for on identity no way of controlling scoping at level of user.
- NIH could not deal with such a system as just doesn't.
- MK: how do you prevent the user accessing anything via TES when passing something that keeps all permissions of the user
- KR: when I signed off on v1.0 it was something i almost different sign off on as unacceptable, I agreed to a compromise, with the understanding that we would immediately begin work on things like scope reduction of the passport and declare v1.1 within a very short time. Between v1.0 and where are here now there has been no movement towards the things that were a prerequisite for signing off on the v1.0 spec at the time.
- KR: never the intention for identity based signing off. We have to do this from a security standpoint and we will do this regardless. We will be downscoping the token, with or without GA4GH, but we cannot continue to pass around the whole bundle of permissions.
- MK: what would your goal be of the next version of passport? Passport as token with all permissions at once in it
- KR: no we never wanted to carry all tokens at once. Walk through the user journey, user provides a log in, saying I am identifying myself and getting access to all my route permissions, next step is to reach into that and grab only thing necessarily for the operation, how do we describe this though knowing what is necessary for the next

operation? Have an approach and will try in dbGAP. You will never pass around all of our permissions for that, but you would pass around a downscoped token that only has the permissions that you need.

- MK: passes one visa in RAS which contains all the permissions, would it be better to split permissions into multiple visas?
- KR: in theory yes and in practice yes, that takes an RSA signature to each one of those and we couldn't afford that. We put everything into one token, the only change in the model from multiple tokens to single with all permissions, you cannot shuffle the deck of visas at the level of the broker. You have to go back to ras to have visa rewritten to issue a subset.
- MK: which API will be used?
- KR: it doesn't exist today as we haven't been able to move the discussion in GA4GH. If we do it ourselves we can do it quickly but getting everyone to agree has been the nightmare for the last couple of years.
- KR: we had to move in small baby steps, shape of the dbGAP visa has a lot to do with taking a small step, our partners such as the Broad were consuming dbGAP authorisation data in non token form prior to this, dbGAP visa structured in a way that made their transition simple. Start using the visas as a substitute. Useful and necessary step to get us to all systems using tokens at all. Compromise that we made in order to get us toward. What we have today was done as a minimum viable step to get everyone on the same page, that was a process. That's why the dbGAP visa is the way it is today.
- MK: why not possible to break into multiple visas, every visa for one dataset or whatever basic unit
- KR: concept of dataset description as URL didn't fit our model, also would have created a large number of RSA signatures that would have made the bulk too high.
- MK: why is it too high, because for some computation they need just a few of the visas
- KR: yes that happens after the downscoping and you still have the large starting point.
- MK: in the beginning the passport and all the visas are downloading from the user endpoint, can be exchanged for a smaller number of them
- KR: here is what missing in the model, you have the idea that everything happens in terms of scoping happens at user and browser at that level. If that was true you would be right, user could be scope before any token leaves but what actually happens is a work flow execution engine it may not be the only next step. We are talking about downstream agents, agents have the same need of downscoping… creates a network not a single case user.
- Model I propose even allows delegate agents to do their work and do their work
- MK: why downscoping cannot be done by delegating a subset of visas? Subset to WES etc, WES may decide may need two TES and each one gets smaller subset of visas
- KR: that is exactly what I am proposing. If user wants to work on datasets A and B, their catalog of resources/permissions user downscope those to A and B and send those off, within WES there is a decision that they want to spend subset A to one side and B to another side.
- MK: then where we are disagreeing?

- KR: agree but why don't we represent everything as separate visas then all doing is dividing up the visa set, that the dbGAP visa, the expression of the permissions themselves does not fit within model proposed in GA4GH v1.0, we reserved custom visa type in order to avoid overhead of so many signatures. We have single visa model. I expect change in the future, discussed for v1.2 is something that moves in that direction and gives us ability to consider splitting the visa. Requires us to recreate a new visa format that breaks our v1.0 visa users. If we are going to break it we should reshape in way we want to go, which is what we are looking at right now. dbGAP needs to experiment now in new ways of looking at a visa. We may be converging to a multi visa model but to do that we would like to move in the compact direction simultaneously.
- TC; nobody likes the monolith visa, so how do we move towards a v1.2 spec that is operable to everyone including Andrew and Max who aren't here.
- MK: one way, we keep specifications as they are and just use special visa, like controlledaccess but putting there some special data. If you just change the type and do something different, it would still fit into original spec. Other way would be to define new standard type of visa which could have the format or content that you propose (multiple datasets) but still backward compatible, third way does not belong to passport, a new special type of token. Which of these ways is the correct one.
- KR: one of the reason even though I contribute to this WG, i do represent NIH interests. I think that what you said is a very ELIXIR view, you make the statement that passports and visas are all about transporting attributes and way NIH uses passports requires us to go for a different type of token. If you want an attribute system then you need some other kind of concept. GA4GH tried to do is to represent a single token system that can do either. If we are not agreed that token system should do either that changes the discussion somewhat as long as still agreed that passport visas do either job then we are using passports and visas to represent authorisations not attributes. From NIH perspective we only have right now we don't preclude possibility that we will use attributes, right now our model is using authorisations.
- MK: ELIXIR and NIH have opposite directions in ELIXIR we have 1000s of home institutions that can specify type of affiliation to Institute translated to visas that need to be collected, most concern is who is the user whereas in NIH you have single authorisation server and all user have account there so you don't care who is the user so you represent access control.
- KR: mostly agree, NIH does not serve as an IDP but do centralise authentication of the users, our authorisations have been gathered at NIH and RAS bottles them up as tokens and send them out, why preservation of original authority so critical, whereas in your model authorisation happens in late in the game, not as concerned about OA. that's good to recognise, positive. Tried to discuss a system to do either.
- Important to not assume that the system does one or the other. The concept of the controlled access visa is handed out. We need to work toward something that can do both.
- Interested in opportunity to create tokens that are no longer bearer tokens. I don't think that bearer tokens are appropriate in a true token model there are cases where bearer tokens fit the application, but in majority of cases people employ bearer tokens when

they really want assigned token and hoping can keep token secret enough to avoid misuse. This is a misuse of bearer token, used as easy to manage

- Assigned token is opposite of bearer token concept, that can be used only by an authorised individual. They require some infrastructure at the AAI level but they are much safer to use if they are sending them out in the world if using for TES. we have rules about encryption, want them always kept in a TLS tunnel. OAuth gets away with bearer tokens as assume simple model where everything is done in browser via TLS, no reason for token to ever hit the disc. But theft of tokens is a major security issue wit OAuth.
- Assigned tokens are no longer so volatile or require such level of protection.
- MK: but passport is not a token, attributes assigned but should not allow access by themselves
- KR: agree that at one point we had a concept of AAI and DURI and DURI also involved DUO. i am talking about authorisation is a lower level discussion, there are tokens that are either assigned or there are bearer tokens. Put assignment at the AAI level, missing piece to allow us to create an ecosystem to describe this.
- DB: token endpoint different from user info endpoint and we can make token endpoint spit something out and we don't need to call it a passport. This thing is the token and different to user endpoint.
- TC: is that a bearer token?
- DB: token endpoint output can be used as a bearer token, but our discussion does not end there. Users can do user info how i want to and get token endpoint to get something to use in headers, which seems to be a real issue.
- KR: such tokens could be used to create assigned tokens that i have called work order tokens in the past, basically a contract that you create with a working agent to do some work on your behalf. I have difficulty when we use bearer tokens to accomplish work, unless work you are doing is entirely trivial or being managed by some sort of a system, if you can't pass those tokens around and expect them to remain secret. We need to contemplate the work being done here and the real challenge is how you do that work with the working tokens, if martin is saying that doesn't sound disagreeable to him, i am happy with passport being passport bearer token and put effort to a derived bearer token used in some way. dbGAP before GA4GH we already had a token system, our token system does exactly that, hands out a bearer token on login and that is used to create a token that gets bound to a compute environment. You have to be logged in to one of those environments in order to exercise it. A specific session if you will in the cloud we get a compute environment from within the cloud and pass that along with authorization token, that is what is required in order to access the data. Assigns a bearer token not to an individual but to a compute environment.
- MK: we are making progress and agreeing now.
- MK:
- KR: shape of that is agreeable but the one thing for our case we still continue to need to see the NIH signature on any token that arrives at the resource server, so just being careful with terminology of exchange and who is the authority that can create this kind of token. Separate ability from separating what you want to work on and authorisation to do

that work which comes back to the data owner. In our case we want to provide some way to have people at the user interface and at the WES level. I would like to subset them in this way.

- MK: so need to define interface of token endpoint so can receive access token and exchange for new type of token and interface between authorisation server and dbGAP so can ask for the content of the new token.
- KR: i think that is what david was proposing.
- DB: yes, exchanging that for token endpoint and getting back token that can be used downstream in applications and maintaining original authority.
- KR:
- MK: important that new token is not called a passport so old token kept as it is
- KR and DB happy with that too.
- DB: I don't want to call it an access token as separate OIDC access token which can be used as an access token but not in OIDC token so DON'T call it an access token even though used in database access.
- MK: workflow token? Attribute token?
- TC: sounds like this doesn't violate the use case of different authorisation servers, kurt presenting the idea from NIH point of view and Martin from distributed view.
- KR: everything to left of authorisation grant follows the attribute model and works well in OAuth. Everything post authorisation has to follow an original authority flow as its the authorisation. Cannot destroy that signature, we need in dbGAP is what RAS represents that very middle point in RAS.

2021-12-02:

Chair:
Attendees - Name (Affiliation): Max Barkley (DNAstack), Kurt Rodarmer (NIH), Martin Kuba (ELIXIR-CZ), David Bernick, Mikael Linden

Apologies: Susan Fairley (GA4GH), Alice Mann (GA4GH)

	Actions Arising	Assigned To	Deadline
1			
2			
3			
4			

5.			
6.			

	Agenda Item	Person/Time
1.		
2.		
3.		
4.		

Minutes

MB: Call for agenda
TC: There's a document with proposals we should review
MK: Last call I was asked to make a proposal. I wrote an email to the AAI list about passport AAI, that didn't get much response
TC: I agree with the ideas you suggested but also lots of the other ideas. How do we resolve this impasse?
MK: What kind of visas are we concerned about? Maybe the data we want to add doesn't fit into current visas at all? Maybe we need an access control visa that has a different structure than other visas. Could have a new construct (maybe called a "Permit") that is clearly an authorization to use a system.
TC: Permit sounds similar to visas. Question to handle multiple has come up from before. Doesn't sound radically different (expect perhaps the format).
MK: Current visas are digitally signed statements about user. Just holding them shouldn't allow you to do anything. Permits would be like tokens that actually grant access to a system.
KR: What kind of visas does EGA need?
MK: AFAIK the DAC store decisions in some format in EGA, visas released through ELIXIR, sent to OIDC client, then the visas end up back at EGA.
KR: And what kind of visas are they?
MK: They are controlled access visas.
KR: So they use the same model as DBGAP?
MK: Probably yes.
KR: So they [EGA and NIH] have a similar model.
MK: Similar, but NIH is putting permissions in a visas.

KR: NIH is putting what I call permissions, but I call them that to side-step use of other terminology in the industry.
MK: Controlled access visa was designed to carry information about DAC decision, not authorization to access the data.
KR: DBGAP has capture of expression of authorization from a DAC, which is essentially what ControlledAccessGrants is doing, but that visa works at a single dataset level, but DBGAP does it at the “permission set” level. What you can is not expressed in the data. It’s implied that the access is read only. It expresses access to the content, not access to the system.
MK: Maybe Mikael Linden can talk more about this?
MK: Is EGA using REMS(sp?)?
ML: Currently EGA has a DB of data access permissions to all datasets. DACs have write access to their datasets.
KR: That’s the DBGAP model.
KR: I think we need to get back to the 1.2 design. MK has suggested alternatives. I responded that the models we’ve looked at have weaknesses in certain areas that can be expanded. Important to realize that when passport started, it was split into AAI and DURI, but in practice they’re difficult to separate. Certainly what archives like NIH and DBGAP focus on is controlled access. Controlled access is a decision rendered on access to data. MK has objected and it sounds like calling it a passport is part of the objection, but if we continue with the analogy passport is container of the visa, and via the analogy visa author is country granting you access not the country that issued your passport.
MK: Passport visa analogy is not a great analogy. You don’t use passport in real life to give to someone else to do something on behalf of you.
KR: Yes, this is something I’ve been complaining about. Passport is nice, but it’s not actually used in a browser. When it’s actually used, people are farming off jobs that may take a month to run. This happens in command line systems, which are not reached via TLS. Browsers are only one part of system that are not being modeled. There are lots of other systems involved acting on a users behalf. It’s not safe to share bearer tokens.
MK: I agree that sharing bearer tokens is not safe. There are solutions like this such as macaroons (sp?). You can restrict them and share them, restrict them, share them again, etc.
KR: The passport model is fine for browsers, but as soon as you move to other kinds of systems you entered a different model. What you are trying to do with the derived token and doing what is called a “power of attorney”. You are giving some other agent limited authority to act on your behalf. It’s a form of delegation that comes with certain restrictions. An important thing is splitting between the token and the message. MB pointed out that a message can be strewn between different parts of an HTTP request, but authority should be attached to the message, not the token.
MK: Macaroons (sp?) are the system used by Google.
KR: Whatever the name, it’s a capabilities system. The model was invented in the 60s. There needs to be a capabilities system on the token and the message.
MK: What do you mean by token and message? Where do you see the difference?
KR: Great question. The token carries some access decision around data you’re requesting. Concept of the message is that you establish a communication channel between two parties

and send messages. In OOP this is method calls. Message carries: (1) indication of protocol, (2) target of protocol, (3) indication of parameters.
KR: You [MK] have been saying we might need multiple tokens for a message. The overall message is the combination of indication of protocol (exact shape of API), additional parameters. It's important in a secure system to sign the whole message to know that it has integrity. We should have a place for all levels of security, but we want a mechanism to ensure there aren't sequence or repeat attacks.
MK: Thanks for the explanation. The problem is on the internet, not all communicating parties can securely store secrets. For example browsers cannot store private keys securely. Otherwise it would be best if every message was signed.
KR: I know what you're saying, but I've designed a system that does this so I know it can be done. If you take into consideration the use cases we've laid out, so far they don't involve mobile devices.
MK: It does because in ELIXIR we have beacon interface that runs in browser, that is just a portal for the user.
KR: I think I tried to say that there are various levels of security you can operate at. NIH for example allows for discovery of data without any bona fides (completely public). Can do that from a mobile device. It seems like there might be some use case for bona fides in discovery but we don't have that yet.
MK: I agree it would be great for all messages to be signed, but they require server-side applications.
KR: Do you use SSH?
MK: Yes
KR: That's an example of a client using signed messages.
MK: There is a confusion about what is meant by client. I mean something like a browser where you're downloading software as a service, and you cannot securely store secrets.
KR: Agreed. You can't use mobile client to do all things in the system.
MK: It's what all users are using these days.
KR: All users are using browsers, but that's not where most work is happening. Most of it is happening in the backends. What we want to solve is Alice wants to run an analysis across several archives. That problem doesn't exist so much at the browser (ignoring discovery for a minute). It happens at compute farms that are computing on the data. That's the number one use case we have to pull together. Discovery is in there too, but not the only problem. It requires delegation.
MK: I agree. We need to design some framework that does all the work between WES and DRS and TES.
TC: So it all comes down to, whether we call it a token or passport or permission, the content of the token. There was something earlier in email about early bound vs. late bound. We're talking about binding a user identifier and a permission to access some data. Do we agree on that piece?
MK: There are multiple layers. In the beginning you have attributes of the user, such as visa. Then you need to take the visas in the passport visas, then you transform them into access control decisions. At that point you have made the decision (Policy Evaluation) and you just need to carry it to the enforcement agent.

TC: So we agree that we're capturing a decision?
MK: Question is if the decision should be in the visa.
TC: There's discussion about format, there's going to be a piece of data. It will have a subject, data reference, and a signature. I think we had agreed on signature.
MK: There are two types of tokens: random values and JWT. [description of pros and cons of each]
TC: These signed pieces of data are statements from owner about data (signed securely).
MB: I think there is some disagreement on the two kinds of evaluation. I think everyone agrees that visas capture an evaluation about who can access data. I think there may be disagreement on where the evaluation to access a system happens.
KR: We've talked about access to data, but not access to systems. I think the model is lacking authorization to a particular instance. For example, I want to model sending a message to Max's system to access data, and he needs to evaluate my license to access his system, which is separate from the license to access content
MB: 5 minutes left; call for next steps?
TC: I think we need to revisit the documents and Martin's proposal. WDYT, Martin?
MK: I think we agree (at least with Kurt), that we need to solve for the backend use case

General discussion of next steps:

- Need to figure out next steps, work on a proposal
- Follow up async
- Should think about larger issues, but no over develop now
- But we shouldn't over-optimize for short term either

2021-11-18: "Multiple passports/visas approach in /token" vs the original proposed "compact custom format".

Chair: Alice Mann
Attendees - Name (Affiliation): Martin Kuba (ELIXIR-CZ), Max Barkley (DNAStack), David Bernick (Broad), Tom Conner (Broad), Kurt Rodarmer (NIH), Heidi Sofia (NIH), Mikael Linden (ELIXIR-FI)

Apologies: Susan Fairley (GA4GH), Andrew Patterson

	Actions Arising	Assigned To	Deadline
1			
2			
3			

4			
5.			
6.			

	Agenda Item	Person/Time
1.		
2.		
3.		
4.		

Minutes

Multiple Passports Proposal - Martin Kuba

- We have to support multiple passports anyway so keep format of visas and pack into multiple passports anyway so the size is kept small
- Proposal here: https://docs.google.com/document/d/11D6t6t6lgvpU9sEWg3f7ZKNqNlQdszJ5P2DFDnC8wXE/edit#heading=h.xz3z2m16d8q4
- What should the parameters for token endpoint be?

KR

- We did a lot of work on this over 2021
- Idea of trying to solve which visas are needed within a passport that derailed our ability to land the visa downscoping issue for passports this year
- Reason we started working on overall passport more compact was because we didn't get traction on the visa downscoping
- Be good to over some of the history of this discussion
- This is the first piece that really affects passports clients a lot
- When you have multiple tokens, burden of knowing which token to use is transferred to the immediate client.
- This is more of an issue of concern to the WES subgroup

- In order to run a proper cross repository work stream, there has to be logic somewhere to distinguish which token to use.
- ----

- MK still possible not to provide parameters to token end point and get everything in one token
- Special parameter for the case for when someone needs the issue token to be small, possible to specify which visas should be. Like downscoping
- KR: how does that get specified and who specifies it?
- My proposal from last year is that the researcher- language translation issue, simple at our level at how visas might be specified by language of specify visas in terms of names and contents not expressible by any researcher. Couldn't present it in the scoping dialogue, researchers won't necessarily understand which tings they need to select or understand what things they are reading
- MK: not job of researcher to specify visas, job of client
- KR: DRS IDs are 100% unintelligible, dont indicate what object you are looking for. Except in case where you might get a hint where domain in full DRS ID. community tried to get rid of as afraid of brittleness of ID if moves from one DRS server to another
- Client may not have any clue what objects DRS ID refers to
- WS take as input, user permissions and opaque object IDs, wont know where object resides or copies of object resides until have requested more information about the object. Request more info about the object will require permission in many cases.
- Chicken and egg problem raised by Seven Bridges
- MK where is mapping from datasets URIs to DRS objects
- KR: visas not all based on URIs
- MK: they are, specification of the visa. Identified the dataset using a URI
- KR: not true
- MB: see here:https://docs.google.com/document/d/1bnnfzcbyZJ8QUkbaYE-wezKsZnS78JZZmDbyJLXGqeg/edit#heading=h.9pxveticgjnz
- MK: spec controlled access grants, value of visa is URL format
- KR: don't leave out custom visas. It would be true if only the GA4GH standard visas are only involved.
- MK: my question is where is the mapping from URLs controlled access grants from DRS object
- KR: DRS at this point understands mapping between DRS ID and visas, problem that I had with what you said is looking for URI that identifies the visa. That works in some cases but not in all cases. Careful about our language around that. Because it's the DRS that understand what object we are talking about and which permissions are necessary. Pursued an avenue to see what DRS coil dbe used to informa client of what visas they need, not same as what URIs identify the visas. Part of it but not the complete answer. We didn't finish that over several months, why we shifted to make everything compact. We need to pick this work up again, as long as DRS remains only identify that

understands mapping between ID and object, avenue to understanding which visas are necessary

- I started this by raising the point that the burden is based on outside clients, so we probably need to involve outside clients, DRS people
- MK: DRS cant be only person understanding the mapping, client needs
- KR: DRS IDs have two main components, authority component and some object ID component. In original spec, authority component expressed as host name to tell you which DRS to call. Extension of DRS 1.1, where that authority obscured and indirected to go to some other place. Resolve authority to particular DRS server, that's all that gives you.
- Almost always the case that the DRS servers associated with visa issuers are the ones that understand the mapping.
- MB: use case we agreed that motivating the doc i linked, usually researchers don't discover objects of interest at a DRS server, normally find out e.g. linked from some other portal to discover data they want. At point when want bytes for these things, they have list of objects they want to pull into an analysis environment somewhere.
- List of objects of interest,figure out what visas to get to figure out which passport i want to send
- MB: we didn’t make it all the way with this. Support multiple visas from token endpoint w/o participation from a researcher who wouldn’t understand things they need, client would work that out from the DRS server, which of these browsers.
- KR: Whether there is a collection of passport tokens or a single token with a collection of visas, it is the same
- DB: We need to match visas to data, then we need that matching info somewhere right?
- MB: suggestion that kurt gave that led to some ideas, idea of just a selection object, here is the stuff that I as a researcher want to access. Maybe this is a more general API that could apply to different kinds of objects, DRS is a namespace inside an object. Imagine other kinds of objects, or just applies to a DRS server could flatten it out more.
- If DRS server can be asked, what passports do I need for this list of objects, don't need to standardised browsing part, standardising after that
- TC: namespace for person who holds the data and then nested ID 1,2,3 which are just strings that make sense to that namespace. In context of DRS 1 those make sense.
- KR: For visas that are atomically identified by URI, the URI is just fine. For visas that contain permission SETs, the URI is not specific enough.
- MB: this could be a way that multiple passports work out what they need. What are hte other ways that clients get what they need in multiple passport world
- KR: multiple passports just using passports where we use visa, it’s just the v1.0 spec
- MB: martin are you wrapping individuals visas so they still have outer passport layer?
- MK: why impression? Because of example of token exchange? My intention you have multiple visas in one passport. Problem is both visas and passports are expressed as JWTs, they are not the same.
- TC:
- DB|: having v1.0 passport spec but coming out as tokens just being wrapped in something but having individuals visas is what is being proposed. Easier to consume is

not a new format but forces us to say this is a token, it will be small. If you want a small token go here. Where can i go for a small token that gives me what i need? Does this multiple token proposal fulfil that in a way that is digestible to everyone?

- MB: i like it if we can figure out how a client knows which of the tokens
- DB: agree, ask of martin give us a written example how it works in a regular DRS flow, i am a user etc. even if we can't capture in the spec, helpful for implementers. Examples even if spec is a bit vague
- TC: we need to go back to each use case and each use case could encompass where did you get each use case. Perhaps it needs to say a little bit more about the dataset. Where did the identifier for that dataset come from, that is a key to where the token comes from
- KR: if we aren't paying attention to WES etc, we are not paying attention to our audience. I think within our purview to establish correct security, we need to make something that works for these workflows. A WES workflow receives some input, doesn't know where tools or data to operate. Partially specified in terms of data to operate in DRS IDs, whatever specifies here is what you run your workflow on. Researcher goes to a catalogue browsing facility and comes up with the definition for work, the work order, passed to the worker. None of these things are running in our browser, or have ability to perform OAuth refresh with the token. Token or tokens that they work on are downstream, don't have same facilities, which is why we pursued a custom token flow. They operate with CL tools and farming jobs off on other servers and compute farms. So unless we design for that, I don't know what we are doing.
- TC: what are we designing for?
- KR: the burden of saying you figure out which token you need for which DRS idea, is going to break the WES logic. Need some other assistance for figuring out which token you need. Multiple tokens i think, not important if individual visas or passports for this part, but do need to figure out which ones to use if hae multiple available. If single token don't need to fixture it out, token contains permissions you need or it doesn't
- MK:
- KR: martin you are saying that you are able to send multiple tokens to each DRS server. Not having them bundled together
- ---
- DB: web servers have maybe 8k limit for all headers from a single request
- KR: There is a per-header limit and a per-request limit
- MB: based on that my stance would be it could still be a valid solution to split up visas into small passports that individually can fit into that header limit, but need to know which ones you need to send and send individually
- DB: why i ask what it looks like with DRS? Requesting for multiple datasets, each has different passport? We need to put that down that this is how we intend it to be used
- KR: what hwen multiple passports authorised for different people?
- TC: what is the use case where that would happen
- KR: not intentional desirable use case, but from security its what we work with. What is the behaviour when someone says here is tom's visa and martin's visa now give me access

- MK: linkedidentities visas for this, these two users are the same, if not then its not secure
- All tokens tied by linkedidentities
- DB: (http://httpd.apache.org/docs/2.4/mod/core.html#limitrequestfieldsize — because i am old, Apache is my go-to — default 8k)
- MK: if dont trust broker then need something to trust, linkedidentities introduced for that
- KR: then only way to get tom+martin passport is if same institution. Should be no way for me to request martin's
- MK: depends who you trust. ELIXIR, have ELIXIR ID released by central broker, ELIXIR AAI. each of them has their own user ids, linked both ways, visa issuer says this user ID which is mine same as ELIXIR ID and ELIXIR AAI says same but in opposite order, this user is from EGA same as ELIXIR ID.
- TC: so 1:1 mapping
- DB: in broad job, terra has hypothetically access to all access tokens and can act on behalf of users
- Terra can act as some sort of agent in the middle
- TC: we work hard people don't get things they are not entitled to.
- TC: in cases where there are multiple researchers needing access to same data, you group them together, how does it work?
- KR: our DRS server evaluates on every request, passport and in case of dbGAP searches for dbGAP visa and evaluates that on every request. Essentially uses linkedidentity if necessary, clearinghouse looking for that even so to make sure visas being used that should be used by the bearer assumed to be authenticated user.
- TC: understanding how they are issued in first place and how people found them. Using standard normal process to get a normal visa and a normal passport. So that brokers and custodians from lots of different organisations can use the same, we need interoperability so that we don't have a lot of players doing their own thing that can't be replicated by other people
- KR: my stance from very beginning is to encourage opaque tokens, understood by the issuer and specific designated participants. Allows the spec to be cleaned and allows for evolution and all sorts of things.
- 1) visas that we come up with right now, tend to be adapted and as we take legacy systems move them toward different values.
- KR: in this document, in original authority, in my attempt to simplify original authority, tried to lay out the stages in access.
- Looking at it in separate steps helps to understand why we may actually be talking about different things
- The whole point of OA is to preserve the signature of the entity that authorised access, for systems that authorises access on the fly. That authorisation if someone comes to you with bona fides, used for on the spot authorisation. If model going to accept anyone with good standing at a good university, then no original authority to preserve to access as authorisation has not happened yet. In the case that we have been trying to protect, RAS issues a token at basically the moment. Authorises a data access request, when

user authenticates token issued at that point, flows from 2 to 5, gateway at step 5 must recognise authority at step 2.
    - To understand original authority, it is useful to visualize a timeline of events:
        1. Request authorization - provide arguments in favor
        2. Arguments are evaluated by authoritative entity
        3. Access is authorized or denied
        4. User requests access based upon authorization
        5. Gateway evaluates authorization and facilitates access
- If arrangement puts step 2 down after step 5, token evaluated by the authority then it may be a different story.
- Is some of this going on?
- TC: i think everyone agrees that authority has to be signed and validated.
- How did you get that signed authority to be created in the first place. Do you go to token endpoint or user endpoint, opaque handle presented?
- Still at what scheme will be workable by all the players?
- MB: how do you know what passport do you even need to get resources? Seems like a difficult problem when thinking about a workflow with 1000s of objects and IDs long, one potential avenue to explore most organisations in practice can reduce various resources to be part of some kind of collection.
- In ASI: called collections. The OA for who can access data, not done at level of individual items, bunch of items part of a study and OA granted access all things in the study
- Justification that people are accessing a small number of studies at once rather than 1000s, that becomes a lot more attractive problem to describe which studies you are asking for.
- KR: and THIS is why the dbGaP visa represents permission SETs.
- MB: is that a universally applicable thing? Does everyone do with a collection or items? We could explore that for how a client finds IDs for studies it requests to
- Can the players agree on the study to ask for rather than a specific visa?
- dbGAP: bundle of studies that you individually have
- KR: object hierarchy what we are referring to here, our designations could go as low as possible, but we work at the level of the container of all of the objects that are under a study. Depends on what consents are there.
- What about at ELIXIR?
- MK: I don’t know.
- MB: actual APIs in DRS, you can get metadata on individual objects. I can’t remember if batch

2021-11-11: Further discussion on compact and proposal of multiple

Chair: Alice Mann

Attendees - Name (Affiliation): Martin Kuba (ELIXIR), Andrew Patterson (UMCCR), Heidi Sofia (NIH), Mikael Linden (ELIXIR-FI), Max Barkley (DNAStack), David Bernick (Broad), Kurt Rodarmer (NIH)

Apologies: Tom Conner (Broad), Susan Fairley (GA4GH)

	Actions Arising	Assigned To	Deadline
1			
2			
3			
4			
5.			
6.			

	Agenda Item	Person/Time
1.	Pick up the discussion from David's emails to aai@ga4gh.org and work on the Summary document	
2.	Is this the stage to report back to DURI WS/Passports group?	
3.		
4.		

Minutes

- MK: not a fan of the new visa format. proposals multiple passports, why not just split the old style visas into multiple passports.
- Token exchange just says it is written as something that is a token that can be anything.
- JSON string that contains multiple tokens
- JWT token with visas inside

- KR: not true that every visa is small. Large passport in v1.1 case is due to including multiple visas also possible in dbGAP that have single visa in passport that passport not significantly larger than passport that multiple bulk in single visa
- MK: WHAT IS IN THat type of visa?
- KR: dbGAP has a permission set in visa, not s single permission. Rather than multiple visas, all the applicable permissions put into single visa in dbGAP. One reason why it's a custom visa. I dont think it's in a final form, visa contents when they are not standard types, when custom types should be considered opaque. Only two parties to understand format are issuer and file consumer to understand the visa
- In v1.0 spec you are right they are small and designed to be
- MB: still curious which part is big? If individual visa still large would compact version of it still be too large?
- KR: majority of size of visa and passport in v1.0 timeframe… v1.0 all familiar, v1.1 nothing more than taking all v1.0 visas emitted in userinfo and wrapping in a standard JWT as David defined in the AAI spec. Single package, but all bulk of NIH case is in the one visa that we do. Far more compact as a representation than if split into multiple visas each with same signature. Single signature applied to multiple permissions. Majority of bits taken up by passport and visa specs is dedicated to non-information carrying bits for human readability, use of URLS are you prohibited to follow in the first place, useful URLS as some guarantee ake sue of name space. Let's focus on human readability as can use translators. More focus on compactness, we certainly can get size down. My v1.1 token is not the largest but not small, my access to dbGAP is around 8k for overall token.
- MB: one visa inside a JWT
- KR: yes
- MK: why do you use URLs that you dont like if visa is big.
- KR: dont use URLS in custom version of hte visa, the NIH part, use URLS in GA4GH portion as insisted upon. I did lobby against but were still put into the spec. So there against my express recommendations from years back
- MK: but in single visa URLs can't take much space?
- KR: overall wasted space in the v1.0 format
- MK: but is the problem for you a single visa, 8kb but large as some non-standard extensions?
- Using URLs in standard part which is not big?
- KR: format chosen for NIH dbGAP visa was dictated largely by the system that wanted to try to read it and used to reading text files. Initially reading information and reconstructing their text file. A lot of petition around the original dbGAP visa was to reconstruct these telemetry files, don't as a compromise, but this is one of the reasons why a compact representation of that is interesting. We don't need v1.2 in order to move to different version of the visa, we could define our own visa type without touching the rest of spec, but rest of spec (AG4GH) does have a very low info:bit ratio
- That's what the v1.2 spec is attempting to address
- DB: The primary reason to keep it small is to pass in authorization headers as a bearer. Just keep that in mind.

- MK: we can keep v1.0 version of visa if you make compact the part you control
- We should not change format of visa as spec already released.
- KR: it depends, from my perspective that GA4GH should keep in mind is that where we are with the passport is we are on a journey, on point in the journey and not at the end. One of the missing parts, no modelling of messages and that is something that i am hoping we are already at this point. I hope we can get to in future, how to incorporate messages into the protocols and make suse of the tokens for this, has a big bearing on how command line tools operate and the safety of tokens as they travel around.
- We want to keep the token small 4k upper limit, martin: what happens if doesn't all fit in 4k?
- Coupe of directions here: first step is saying cannot exceed 4k, then we define either an escape mechanism when it cannot fit in 4k, we have to represent in a different way and/or the permission set are good candidates for that permissions et to be pre-defined with permission set and reduce number of permissions in the token. Having that the permission set represented withhe super set ID. difficult that we have idea that these things are open specifications, andrew raised a point in document that not necessarily a good idea or practical idea.
- MK: for the visa must be understood only by the broker, issuer and consumer of the visa, nto entirely correct. Clients may need to understand the visa as well. It should not be some opaque string that cannot be passed.
- KR: correct. It is not intended for client or consumer, intended to indicate from the authority indication to the final recipient of what permissions are available.
- MK: cannot presume how the visas will be used
- KR: is it useful to clarify that are two different concepts of visa in GA4GH. When project started a lot of the concept of the visa and motivation for the visa. At least for the passport itself is to carry credentials and bone fides, seen as a way for reducing the barrier for researchers so didn't have to request and receive authorization for obtaining data. Important to distinguish from security standpoint between bone fide and authorisation. This gets at the heart of the questions over OA and chain of trust. Someone at my door with bone fides, most important source of authority is me in that case, i will look at the bone fide and i need to understand them and know whether i am going to make an authorisation on the spot.
- MK: what do you mean by me?
- KR: i am the resource server and I am saying someone comes knocking at my door with token with no authorisation but contains credentials, i am gong to render an authorisation decision at that point, quite possible in that case that chain of trust flow is the most appropriate.
- In the case where authorisation is issued by authoritative source, once authorisation issued, the authority behind it has to be preserved, mortication behind OA flow, reason why not simple tokens. Authorisation is basically your pass to getting to the information that you are seeking. Very important to understand the difference between those two types of visas. Big difference between bone fide visas that contain my request for data or mt credentials for asking for data and pre-authorisation that says the bearer of this token has authorization and you will let them through, two wildly different things.

- Pre authorisation early bound authority where bone fides are late bound.
- When you have bone fides you may be right and maybe they need to be open and everyone needs to be interpreting that. They have different functions. Both assertions can be right but depends heavily on your viewpoint.
- Max from his standpoint dealing with people asking for access, my take on it? When doing so different decision in the case of say an NIH system where NIH has said the bearer of this token when they come to your door has access to this information under our authority. That's what you have to do, the decision has already been rendered. Thats why the mechanism we talk about has to support that capability. We are looking for something that supports both.
- MK: if i understand correctly, the difference is between controlled access type and the other types of visas, is this correct?
- KR: yes if we dont talk about custom visas, of open pre-defined types, controlled access pre authorization, early bound authorisation.
- AP: because binding between person issuing controlled access and person giving data is considered a tight binding, agreement to format is between those two parties inherently. They could have a custom format. Could agree to agree on more general visas and more specific controlled access is agreement between two parties, not 100% necessary for a standard body to get in way of that.
- KR: issue where rather than listing permission by permission, if i recognise that andrew has a lot of access into dbGAP and his tokens or passport >4k, I can create for special case and represent that in a special case. If everyone trying to interpret the vias then will run into problems and won't know how to interpret that.
- AP: could maintain all current visa formats. Other than boiler plate visa stuff, you could literally have a custom visa just for the nih any optimisations in the JSON. it could be a compact visa format and that would be a standard for NIH to have between NIH systems for NIH controlled access
- KR: yes I think that's true
- AP: agree for this NIH thing there is a more compact representation done. But also agree there is an overhead of the way visas are done in terms of large signatures and base64 encoded things inside bse64 encoded things. Not end of world but losing 20% of storage for that, could just take that hit?
- We were attempting to get rid of all wasted space with compact 4k
- KR: remind that v1.2 if v1.1 spec only taking v1.0 and bundle into single token. V1.2 taking v1.1 and compressing into single token so doesn't take up so much space. Why do we need v1.2 if can recode visas and be happy, i think there may be a path there certainly. NIH has not accepted v1.1 as a standard. V1.1 took JWT standard and wrap up individuals visas in JWT. if not doing v1.2 then must be v1.1.
- MK: V1.0 has custom visa types for NIH
- KR: yes we did that but v1.0 didn't define a passport, read wording passport.
- MK: it says that passport is access token + response from userinfo
- KR: no it doesn't, spec is clear about the visas and formatted and they are JWTs, def of passport is obtained from userinfo within a particular claim. This is something that post v1.0 we went over carefully and reason that NIH tried to get GA4GH to go further and

introduce the definition of passport as the token. It was widely interpreted and originally intended as v1.0 and in prev1.0 timeframe that it would be a access token. OAuth access token does not support original authority so it was never became a passport in v1.0 time frame.

- MK: access token cannot include much info as becomes large and has to be passed in URL, limited to 4k. Not possible to include visas into access tokens, reason why they are in userinfo instead of access token.
- MK: i did implementation for ELIXIR, put in access token and not working.
- MB: From the top of the spec doc: "A logical concept that includes a Passport Bearer Token along with any Passport Visas that may be acquired by making /userinfo calls to the Passport Broker using the Passport Bearer Token." and "The Passport is a logical concept in v1.0, but MAY have a unified byte encoding (i.e. no longer just be a concept) in future revisions of this specification. This is beyond the scope of this v1.0 specification."
- KR: basic problem that OIDC access token obtained from system, access token to userinfo endpoint and it was used to obtain a collection of visas rather than a token, this implied that what was passed around as the passport was the access token, access token has acces authority and does not support carrying visas passports. Left intentionally vaguely defined.
- V1.1 timeframe passport was a token solidified, token passed around, single token as david mentioned keep it pass and pass as bearer token in the header. The v1.0 spec of the passport does not describe a token that carried embedded visas or carries visas as embedded tokens very well.
- MB: From the top of the spec doc: "A logical concept that includes a Passport Bearer Token along with any Passport Visas that may be acquired by making /userinfo calls to the Passport Broker using the Passport Bearer Token." and "The Passport is a logical concept in v1.0, but MAY have a unified byte encoding (i.e. no longer just be a concept) in future revisions of this specification. This is beyond the scope of this v1.0 specification."
- HS: address the definition of passport from Max
- MB: passport as conceptual/logical concept
- KR: that is exactly what i was trying to say, logical concept within a spec rather than a concrete thing. If contrast to def of visas, they are well defined, JWTs, that have these properties that has this logical spec, clearly states where you get that information. But doesnt say what a passport is yet, still left as logical construct.
- Exactly what i brought back to GA4GH, and raised and NIH moved ahead with the idea. Passport could be defined and in what was JWT that otherwise just had these visas as Craig described them.
- That;s what the v1.0 passport is.
- MK: no disagreement about this, but if say passport is a token it can be large, then somehow pass a large passport around
- KR: yes that’s right, almost no issues at level we are looking at that are single variable equations, if we solve for one variable we don’t solve the overall problem. This is a multivariable equation all of them simultaneously. I recommended that we move away

from passing tokens in headers for new API. APIs have opportunity to define their own protocol and should not be hampered by trying to cram a token into that header. If go direction of DRS they start to be able to model messages and messages can be formalised and contain multiple tokens and parameters.

- About the message not the OAuth token. So far in GA4GH we have left it out of the picture, we have focused on how to slip token into a header that is not connected to the message in any way.
- Message:
- No control over replay or sequence attacks that is basic communication technology that OAuth have been ignoring entirely. DRS has said here is how you address DRS with a JSON object posted to the server. Starting point formalising around messages. Oauth token becomes a part of a message.
- ---
- DB: “Which is a requirement for downstream things like DRS” — sorry things are loud here with kids home.
- We should move away from passing tokens in headers then not defined by single tokens, not have to embed visas in another token in order to pass in header as a single token. Whole idea to embed visa in another JWT is exactly because trying to do everything with single token. Represents user authorisations (...), becomes very difficult to manage in a single token world
- Idea of embedded visa came from being able to preserve original authorisation tokens from multiple source from a single token, caused us to look for a two level thing. Double case 64 encoding all of that
- AP: first element in payload an auth array, client puts in the visas themselves rather than passports themselves or still put in a signed JWT passport. Passport from two different brokers, take all visas from brokers and put in auth array and post to an endpoint.
- KR: …. back to single token where each representations its own authorisations, client embed auth tokens that are necessary. **…**
- MB: nothing about http headers that prevents you from sending multiple tokens. Could decide we want a GA4Gh header provided they all fit in headers. Size predominant issue if solved fro size you could put as many headers as you want and be validated separately from bearer
- KR: true but lose ability to bundle authorisation with the message so you cant secure the message itself. If serious need to secure message itself. 3 levels, session, message, and authority behind ability to send message.
- MB: in multiple token message can replace bearer token, sign your request but still send passport visas in header
- KR: message itself goes in header?
- MB: no but request signing another common technique for http that is orthogonal to how you send messages. Most request signing includes header
- AP: whatever set ..
- KR: quite true.

- MB: if people think that is important for their resource servers, could be orthogonal. DRS API requires bearer token or recognised key and GA4GH headers. Outside scope of passport and AAI be up to those APIs to decide
- KR: disagree, mandate of this group is to establish te ability to communicate securely so rest of GA4GH can build on with total confidence so they don't need to worry about it anymore. We need to get this as close to right as we possibly can.
- If we go down multiple paths then what is the client to do if the foundation bifurcates? How do you manage that.
- All of these struggles and discussions useful to get on the same page. Ways to improve the security.
- We need to steer this to creating signed tokens. Token refresh a big problem. All can be addressed by taking a different approach by sending messages where tokens useful on their own, have to be used with a message and the message is bound to an assigned user. Id like to eventually address but address step by step
- MK: we still dont know what GA4GH passport should be used for. Digital passport for his or her access. Passport should be information about the user, freely accessible and auth decision made in the system which needs to make the decision, not done at visa issuer nad passed as a visa until gets to some consumer. Decision should be done at the consumer based on the visa
- KR: late binding and provenly insecure
- MK: why?
- KR: idea behind late binding is to support cases where you have excess authority and that’s the short version the longer version I can explain give more time.
- Will send something written on this to the group.
- Old split in the community between authority based and identity based security.
- AP: explicit statement of visa saying this person allowed this dataset access better than someone turning up at end point that this point is researcher and works at this organisation and having RS make decision on those, these are vague permissions? Is that too much power? Better than controlled access visa?
- KR: yes agree in spirit but is a place for vague, not saying that should be disregarded, i am saying this not the default.
- This is split behind ELIXIR and NIH view, ELIXIR search as have researcher status.
- KR: very black and white, we are trying to solve problems that have completely different requirements.
-

2021-11-04: v1.2 plain text format

Chair: Alice Mann
Attendees - Name (Affiliation): Martin Kuba (ELIXIR), Andrew Patterson (UMCCR), Heidi Sofia (NIH), Tom Conner (Broad)

Apologies: Susan Fairely, Max Barkley

	Actions Arising	Assigned To	Deadline
1			
2			
3			
4			
5.			
6.			

	Agenda Item	Person/Time
1.	Details of what the plain text format of v1.2 actually looks like - Once we have details here we can send this out for comment from the broader passports group.	
2.		
3.		
4.		

Minutes

- AP: Max has convinced me that the migration story is better than in that document,
- Only migration story has to do with visa issuers. How do you do backwards compatibility for the visa issuers, need to issue two different visas unless need to do clear cut overpoint.
- Is there some sort of complete interoperability between the two visas formats
- Is there a real complete compatibility between the two visas, then issuers are in a pickle. can represent in one way and not in the new visas.
- The visa issuers are so downstream how can they possibly know which visas they are producing? There will be a transition where they need to generate both types of visas.

- TC: would it send both in all cases or able to send both depending on what was requested. Would there be a version field in the new /endpoint?
- AP: possibly as a bespoke kind of communication between broker and visa issuer. Is the things we can represent in plain text visas 100% the same as what we can represent in v1.0 visas. My claim at the moment v1.2 plain text visa format is as expressive as v1.0. v1. 0 is infinitely expressive as a JSON object and entirely expressive. Going back to martin's point we may end up reinventing JSON to get equivalent level of expressivity, which seems silly. Maybe better to encode current visas in more compact visas rather than different visas.
- TC: are the use cases expressed in v1.0, are they edge cases or are they unusual, is it okay to leave that functionality behind.
- MK: I see another potential problem, with compact visas, sare some size by putting multiple datasets in one visa so they cannot be separated later. In v1.0 each dataset is in one separate visa.
- AP: Kurt's idea seems to be downscoping mechanism to be developed for that. That you would in some way deal with that. You can't untangle that visa can you, broker can give you back a different subset of visas but can't give you back a visa with less datasets in it.
- TC: the issuer could re-issue? It always comes back to the use cases.
- MK: are we talking about all types of visas or just controlled access grants? Do we need to put other types of visas into token that is issued by token endpoint. Other visas like linkedidentities, researcher identity. They are more about the person. In ELIXIR I already have visas of this type without access to dataset.
- TC: not sure either.
- MK: important for use case where you have researcher from some organisation can be millions of them, need some portal or web publication which first checks its a researcher and able to search for datasets that are available publicly.
- TC: what sorts of principles are identified in the visas, is it only persons. Does it accommodate groups instead of individuals and should it?
- AP: i think that was one of the clearer use cases that I'd thought had come out, was the concept of a researcher status. For us in australia this would be I've done a data handling course at Australian Genomics and by doing this course I get this badge in my visa for this, it's not a I work at this Institute, and that for instance wouldn't give you access to specific data but give you more powerful searching. The system would at least you know don't do identification and re-anonymise the data.
- Affiliationandrole: one of richest, it would have sub clauses to say you've been authorized into this controlled access dataset but only whilst you main an affiliation at the Broad., you could express this richly.
- TC: in practice it may or not be important to have the passport express the person at Broad. You could lose their right to log in.
- AP: where you've got the multi institute log ins.
- MK: yes i have multiple affiliations, institutes, google and github.
- AP: when you are doing controlled access grants they are given to your ELIXIR identity
- MK: yes, affiliations are given by what identities you used in the last year if i remember correctly. If you quit some institute you may still have this. It's not possible to distinguish

if somebody quits the Institute but never logs in from that Institute but is still employed there.

- TC: what are the typical expiration dates
- MK: usually I did one year. For the visa about agreeing with the rules, 100 years as it's forever.
- AP: visas would be long but passports would be a day or two, an individual passport would be long enough to submit to a realistic bioinformatics job 3 days and otherwise you'd have to come back, and not used in practice yet.
- MK: no one was really thinning about expiration times.
- What is the NIH time?
- (we can ask Kurt)
- TC: revocation as well, if there is a reasonable revocation story, does that handle it? It's very different use cases if passport meant to launch one job analysis and passport gives you access for a year or two doing multiple jobs with this data. Both are valid. Are there rules in the spec about it or not.
- AP: we should have a few more discussions in spec about how people envisage how
- Is there 100 brokers in the US or is there one broker in the US? Are visas or passports long or short lived?
- We only have high level use cases at the moment.
- AP: explanations by example would be great, ecosystems exist currently that use visas.
- NIH tighter federation, a prior registration, mutual TLS, broker and DACs are tightly linked
- ELIXIR much looser federation.

David Bernick joins the call

- Visas that don't carry access to a particular dataset but affiltiationandrole, these would not be accommodated currently in the v1.2 spec
- DB: good question, I don't know
- MK: if you want to carry all the visas everywhere, every system you want to have a logic.
- AP: you could punt some of that through the broker. We haven't specified how the broker speaks to the visas issuer.
- TC: at least two different points of checking. The mapping expressed in the visa and passport group of visas, independent of that the person could have their access rights revoke. I don't want to write it in the v1.2 spec, i think we should move forward with compact, solve the problems that Andrew had identified. As we start to get that more real, then others that think we are missing huge use cases can have the opportunity to say so.
- Let's go forward with what we have
- Issuer: parent node in the tree, you already had the solution to that.
- What do you think? Full speed ahead? Set aside the question of particular features that may be lost in v1.2, we can discuss those again next week, week after or if people not concerned.

- DB: full speed ahead always. Yes I am fine with that. If we get wrapped up in the edge cases, we will never move forward
- AP: I agree with true edge cases, super sophisticated. I worry about things that are currently in ELIXIR visas. I don't feel we can go forward otherwise ELIXIR won't go to something new until they get clarity.
- We have a clear understanding of what visas NIH have in use, their own bespoke one for controlled access, with dbGAP info. Kurt seems happy with a bespoke plain text thing. Want to get a feel for what is the scope of visas currently in ELIXIR.
- AP: to institute a small breaking change and then allow both, to allow any style visas. Passports can now contain both old style visas and new visas and people can choose to put into the passport either and that way you could represent the things that are representable in compact visas and if not got around to specing affiliation, people could put in the older things. In terms of moving forward with a plain style spec. Breaking change is to say people who inspect passports, just CHs, things that come in are now not an array of strings, are array of strings or JSON objects. This is a minor thing as a programmer.
- MK: i don't know how many systems use the passports. I don't know how many how many are actually used.
- MK: about 30 services in ELIXIR who are consuming the visas. Not everybody is in this room.
- DB: what is the breaking change? I think the userinfo all stays the same?
- AP: yes they still go to userinfo but the passport they get back now contains both style visas. Visas are in there are either JWT visas or JSON visas, a small programming change, you need to loop through strings or objects rather than strings. But it is a change that would break my code for instance. Two paths you could go, migration story, nothing changes for anything else in the system visa issuers now need to issue both, theoretically have the migration period.
- DB: yes ok agree
- AP: or no migration story, then issue what works for them at the time.
- DB: it's that, if i am advertising as a v1.0 broker, you don't have to embrace new token only methodology. If we advertise as v2.0 you could support that new one, but downstream consumer has to know what they are getting into and what they are consuming.
- AP: in that world the visa issuer needs to support two. Visa issuer now has some complex thing need to express in v1.0 that they can't express in v1.2. That would continue forever. Not in a position to know downstream how they will be used in v1.0 or v1.2 world. So visa issuer having to do both.
- DB: is that ok? In oauth world, it didn't catch on and oauth 2.0 came out and broke it and OIDC came out, non are compatible but all evolutions of the concept but now widely used as OIDC. it is ok if they broke along the way.
- MK: we have EGA which is something like NIH and it took us about two years to provide as simple update, call URL and return all visas they have. Took us two years to implement. I still have the feeling not good to create a new version of spec when there is

a big chance that the next version will be another breaking change. We are still not decided how it will work.

- TC: if we make this change to make this change, a hypothetical 2 doesn't necessarily break a v1.2.
- MK: I don't think the AAI would work by passing around all visas all the time. There must be some downscoping mechanisms, this problem is solved in high energy physics community by macaroons, some tokens that still keep original authority but are downscoped and limiting power given by the original token. Suppose we will need something similar?
- DB: sounds very interesting.
- AP: at some level some of the new visa format i don't know how it gets driven other than by existing implementations. What NIH and ELIXIR want to do in this space, even if it takes EGA 2 years to make any change at all, feels like it needs to be dealt with. No idea how hard it is to get bits updated. That 100% has to be the migration story.
- DB: I agree with that, that those two big drivers are the ones telling us what we can and can't do, which is reasonable as things that aren't use die. How do we bridge those two world views, moving to v1.3 with macaroons sounds interesting. We need to stabilise what a passport is. I know we can't just constantly change, I don't think NIH is any faster. RAS process has been around for 4 years already.
- ---
- MK: still not convinced we want to issue the token in passport.
- DB: due to original authority maintained. Passing it to authentication through a POST was not acceptable had to be in header. NIH was fine with big passport and being in a POST. other entities were not.
- AP: NIH restriction is that they need passport to be a bearer
- DB: maintain OA
- AP: but also be a bearer and not communicated by a call back
- ….
- MK: if NIH does not need this passport to be small, can we not have token as token endpoint but as original visa format.
- AP: Kurt disagree with needing bearers in tokens, drive came from other people. Deploy in AWS, has a built in bearer in header that i may want to use. There was a general feeling in htsget people that a header based smaller bearer token they could see how that would seamlessly fit into DRS etc.
- DB: general consensus
- DB:
- TC: keep spec-ing v1.2 out, once more explicit, good review and ratification of this and seeing if there's an adoption. People not using passports as too big or reasons you just summarised. This will expand the ecosystem by getting this compact passport.
- MK: new format less expressive than original then it's limiting.
- DB: do not disagree with that. Measure to get people moving forward. Not everyone could adopt it.

Actions

- We need to know who is using passports currently, who is consuming it?
- We need details on the ecosystems, NIH and ELIXIR, what sort of visas are being used?

2021-10-28: Migration story, versions and v1.2 plain text format

Chair: Alice Mann
Attendees - Name (Affiliation): Alice Mann (GA4GH), Kurt Rodarmer (NIH), Mikael Linden (ELIXIR-FI), Max Barkley (DNAstack), Andrew Patterson (UMCCR), Heidi Sofia (NIH), Tom Conner (Broad)

Apologies: Susan Fairely (GA4GH), Martin Kuba, David Bernick (Broad)

	Actions Arising	Assigned To	Deadline
1	Email Andrew if you want to flesh out details for the architectural decision register		
2	Add to the migration story in this document (see below) - we will work on methods for backwards compatibility between v1.0 and v1.2	All	
3	Continue on with the Summary v1.2 defining the details of v1.2 proposals		
4	Next week we will discuss the plain text details		
5.			
6.			

	Agenda Item	Person/Time

1.	Migration Story? Would this be in the spec? We haven't decided where this will live yet, so it is remaining in this document for now. Please edit!	
2.	Token endpoint to the decision register?	
3.		
4.		

Feedback from Jeremy on versions and product approval:

- Re **Version Control:** if the group decides this would become a Passports v2.0, it would need product approval. If a 1.2 is sufficient then it would not need to go back to product approval.
- The main difference between whether to increment to 2.0 or 1.2 is if there are any **breaking changes** in the new version of the spec.
- If there is a new way of doing things, such that the old way of doing things (in 1.0) is not possible anymore, this is a breaking change and needs to go through product approval.
- If the 4k passport structure is incompatible with the original passport structure, this is evidence of a breaking change.

Passport/visa Format Migration story (i.e compact visas):

I think there are 4 main players in the ecosystem -

1. Visa producers (across a variety of components)
2. Brokers
3. Clients
4. Clearing houses (also possibly a variety of components).

Of these - it is true that

- Clients are essentially agnostic to the format of the passports/visas - but need a story about knowing if there is a particular flavour of flow that they should be using (*so they have a different AAI compatibility/migration story that is not passport/visa and should be dealt with elsewhere*)
- Brokers are generally agnostic of the visa format, but still possibly need to provide support for the migration stories of visa producers (do I put all of them in, or do I know for each client/clearing houses whether they want old visas or new visas)
- Clearing houses need to be able to recognise visas - and so are sensitive to the format of the visas. However, there is also a reasonably tight coupling between the visa producers and clearing houses that want to accept the visas. I.e. a clearing house is already only recognising a **subset** of the visas in any given passport (those with known issuers/signatures) - so there can also be an a prior agreement as to the actual visa format too (jws v plain text (or indeed non standard)). So I think there is hope that you could possibly retrofit the visa array structure so that it holds a variety of visa formats - thereby making the migration story for clearing houses going forward relatively

straightforward (only recognise those visas that are both from an issuer you expect *and* in a format you expect). However there is definitely a one-off breaking change to introduce this - as currently clearing houses are expecting visas to appear in this array

```
 "ga4gh_passport_v1" : [
    <Passport Visa>,
    <Passport Visa (if more than one)>,
      …
  ]
```

Where <Passport Visa> is definitely a single string - “Encoded as a GA4GH Visa in JWS Compact Serialization string format”.
So we could change the definition of the array to be
“An array of strings or nested objects representing visas” thereby allowing

```
"ga4gh_passport_v1" : [
    “eyasdadsasdadadadadadsadada”,
    {
      "v": "c:8XZF4195109CIIERC35P577HAM et:1665130508
iu:https://nagim.dev/p/wjaha-ppqrg-10000 iv:39a277efae72236a",
      "k": "rfc8032-7.1-test1",
      "s": "FWAYv00igGtQVPv6GLDCX5inGSWi-IaUldDw"
    },
    “eylkop235o23k5op234k2o43k2o4p24”,
      …
  ]
```

This is unequivocally a breaking change.
Alternatively we could say that any compact representation is just an alternative more compact serialisation that is still **represented by a single string** - that would require very little change - I would still say breaking change - but others may differ. This is more along the lines of approaches Martin has considered (compression, elliptic curve etc but all in some base64 single string)

Another alternative would be to introduce a “ga4gh_passport_v2”: [ ] array - which would allow older clearing houses to work without change - but would require every visa to be represented twice during the period where it is not clear to the broker what type of visas the downstream clearing houses need (possibly can we make it clear to brokers what the client wants - alternative openid scope “ga4gh_passport_v2”??)

- Visa producers need to provide signed immutable artifacts - so are obviously inherently sensitive to the actual format of those artifacts - they have to produce them! During a migration period it would be possible for them to provide visas in multiple formats to the brokers - all signed - though this becomes harder if there is not 1:1 semantic equivalence between the new and old visa format. For instance, one of the pluses to the new plain text visas is the possibility of merging multiple assertions into a single plain text string -

but we have not yet established how that could be fully equivalent to the very rich (and extensible) JSON structures of 1.0 visas - that makes each assertion (of a controlled data set) in an individual separate visa. I.e. can we make "c:abc c:def c:efg" semantically equivalent to [{ "type": "ControlledAccessGrants", "value": "abc", "source": "https://ega-archive.org/dacs/EGAC00001000205", "by": "dac" }, { "type": "ControlledAccessGrants", "value": "def", "source": "https://ega-archive.org/dacs/EGAC00001000205", "by": "dac" }, { "type": "ControlledAccessGrants", "value": "efg", "source": "https://ega-archive.org/dacs/EGAC00001000205", "by": "dac" }

Minutes continued

- AP: based on the above is this a breaking change.
- Clients non-breaking, brokers non-breaking, clearinghouses a breaking change.
- Big change at visa producer end, but mitigate by producing multiple visas.
- Mainly at the Clearinghouse, how can you replace the strings in an array with non-strings without it being a breaking change.
- MB: it's like Http2 versus http1. Clients and servers that didn't break because http2 was added as could negotiate which one they could use, they could fall back to 1. If that's a situation we can replicate here, where we can fall back to using Passports v1.0 then you could justify that it's not a breaking change.
- AP: agree but problem is a federated system, easy in negotiation between client and server. Here we have a large federated system, not sure if client and broker are in position to make judgements on a downstream CH. I can't picture in my head what that negotiation would involve.
- MB: just between client and CH maybe? CH needs to understand what sort of token it processes. If you could find out as a client, you could ask before. Get the 4k token and send or not or do whatever you were doing before.
- AP: true
- MB: difficult as RS explaining what kind of visas/passports looking for, we had already decided to punt. Goes to point to expressing what sort of issues you need. How much does that open a pandora's box for versions or something simple that could be added. Hack into /userinfo and find out what version of passports you support.
- KR: I think the thing I am hearing that doesn't sound entirely correct is that a client would be in a position to determine what type of token to send downstream for a single CH. but how does that address a passport downstream that would end up at multiple CHs.
- AP: it could be such a large set of distributed CHs in a federated, how can you go the negotiation like http, as you client as no prior knowledge of what the CH is going to understand.
- AP: from a CH perspective, inherent step where they discard visas they don't understand, concept of throwing out things you don't know or understand is already there in CH. so simple to say instead of strings, have a variety of passports format so throw out if you have passports you don't understand, so relatively small breaking change but still one.

- MB: possibility of letting people use the 4k to hit the userinfo endpoint, so you could fall back to using it same as v1.0.
- KR: We did. Semantics of the various tokens, OIDC access token going to /userinfo whereas the Passport does not. That's a specific unidirectional downgrading of the authorities conveyed in a token. Cannot use the passport to go back to /userinfo.
- MB: is there any possible augmentation to make that acceptable? Send to userinfo and just showed you things in your token, like v1.0, in old JWT format
- KR: david suggested we dedicated an endpoint to passport manipulations and format conversions in the future. Where we would need to expand a passport out in order to have it processed by a clearinghouse and we may drop the permissions to filter it down. We need at least one endpoint to manipulate passports and this would be one way to approach that, i don't think you could make a point that overloading an existing endpoint buys you anything.
- MB: it gets you backwards compatibility at some level. It isn't a mechanism for changing the scope of what permissions are in the token.
- KR: that BW compatibility translation intro the new upcoming release, someone has to add it. So they are aware of how to deal with future formats that they don't understand, so they could a different endpoint. They wouldn't know to call an existing endpoint.
- MB: CH would not need to know to do this translation. To them it;'s a JWT (token) and they can send to userinfo endpoint they know in v1.0 and from CH perspective everything would look the same, yes the broker would need to do extra work to have backwards compatible interface.
- AP: breaking change then at Broker and visa issuer.
- KR: how many CH exist in the world that we are worried about breaking? We are talking about this as if they are 1000s. People in the room are the only ones to do this.
- AP: not how standards work.
- KR: but we don't have standard input and standard output. In the NIH model, the token starts it's life at visa issuer/broker tied down in RAS and ends at the CH. those two are tightly paired. The CH has to know how to deal with tokens. No valid expectation that CH are entirely generic and would never have to be updated.
- AP: I agree that this process is not a standards standard process.
- KR: we pretended discussion is only v1.0 and v1.2 but a large proportion of the ecosystem also uses 1.1 that was not officially adopted.
- KR: at least half has already had to manage the v1.0 and v1.1 issue, NIH has deprecated v1.0 and will not support v1.0, so if you work with NIH you use v1.0. For someone to pretend as we have something v1.0, it was a toe in the water test that isn't going to survive.
- AP: potentially NIH's ecosystem is less true federation, ELIXIR may not be in a position to declare all nodes of their federation to go from v1.0 to v1.2. NIH may be able to have control to have all the software pieces. ELIXIR is federation across different countries.
- ML: 30 relying services consuming Passports, certainly an issue if you want to support several versions at the same time.
- AP; It would not be trivial even with willpower to put a cut off date to change?

- ML: if we want to issue a standard period for earlier version of passports, think it is doable. We need to discuss internally.
- TC: this is a potential breaking change. Whether or not a practical showstopping problem to maintain two separate channels as a migration path, is it impossible or impractical to have older and newer channels as active options. Whilst the older mechanism phased out. Cost to maintaining a older mechanism but I am not understanding as to whether this is viewed as impossible. Can we go forward with this even if it's a breaking change.
- AP: imagine 6 month window where doing a migration, visa issuers need to issue visas in both formats simultaneously, semantic equivalence issue between new and old style visas. I'm not sure we have done enough work on plain text format to do that, old JSON visas allow nested JSON structures, array of conditions in it, complex nested data structure. What does it look like as a plain text visa, you'd be asking visa issuers to issue two visas in semantically different formats. Now issuing two sets of visas, now arrive at the broker. Broker needs to make a decision if broker wants new or old style visas. Now doubled size of passport not shrunk the size of the passport.
- MB: we were never putting olds tyle visas in passport access token. V1.1 you can get a jWT with visas in it, a separate JWT, still separate for token exchange to get the 4K one. Do we all agree that breaking change doesn't mean you need to implement something different? If one party in system uses v1.0 and another uses v1.2, they can still interoperate falling back they can understand. In order to accomplish this one party need to do a lot more work? Okay if v1.2 if a broker needs to work hard to fall back to v1.0 as long as still maintains the security properties we want.
- MB: explicitly exchange that the client has to do to get a 4k token. Client doesn't understand v1.2, still has passport access token. Is visa issuers don't understand new compact visas, broker doesn't put in 4k tokens and only expose in v1.0 format. Only one sticking point what is one broker nested behind several RS understand it?
- But it doesn't seem insurmountable to me. Or we extend the process of creating this version?
- I would personally much rather brainstorm graceful fallbacks.
- AP: yes agree with that. Maybe thinking v1.1 a base not v1.0 for my above migration story.
- MB: even in v1.1 it's in a different JWT. I don't think anyone in practice supporting a fallback but could still have a graceful fallback in those two versions. As long as different tokens there is a way to fallback.
- AP: the main compatibility is then a migration story of what do visa issuers do if they want to support both. Produce v1.0 and moving onto v1.2? For a while will have to do both to brokers to allow brokers do the heavy lifting of the migration story. V1.0 are much richer semantics.
- MB: if said allowed to use EC instead of RSA, would people consider that a breaking change? My opinion a little bit but still getting away with this being a 1.X version. Sneak that into v1.2 as well as gives a potential upgrade path, using EC instead of RSA, don't need duplicate public keys in different formats. That can be part of the migration story.
- AP: I agree with that. If I am a CH I am already throwing out

- KR: I would have to differ. I think the concept of being a non-breaking or slip it through as not doing proper validation. .. this would break out validation before getting to signature checking. We ignore visas but not passports that were being sent to us.
- AP: suggestion for EC on visas not on the passports.
- KR: we check things out carefully, and if there are conditions under which we ignore visas where they look like they are valid. When do not conform to spec to they look like suspicious payloads
- MB: if we propose hypothetically v1.2 but like v1.1, but visas only using EC not RSA. would you consider that as v1.2 or is that a v2.0 in your mind?
- KR: that is fine as long as RSA continues to be acceptable. V1.2 discussions are embracing EC, just trying to say if you try in general case to slip this by under impression that all that is happening is JWT library and probably ok.
- AP: i don't think we were meaning that. Reasonably trivial if a standard feature
- KR: I agree with that.
- MB: I think all in agreement here. Addition of EC to the v1.0 format of visas should be included in v1.2, makes migration story nicer as initially transition visas to use EC and then write in two different formats with the same key, make transition period easier for implementers.
- AM:
- MB: discussion, happy to build up from there as to what the migration story. Do we want to keep that discussion in this doc. Will add some comments or suggestions to start discussions about what mitigations. Visa producer story still needs to be told.

V1.2 summary document outstanding points

- Resolve basic Oauth discussion
- AP: what do you think about some of my clients, that have no secret. They don't authenticate themselves as a client to OIDC endpoint.
- MB: think it's just any method that is allowed recommended. All the same rules that apply.
- AP:

New Passport spec discussion

- AP: plain text spec is not spec'd yet. 100% not semantically expressible as what is in existing passport spec
- TC: David had in mind, take the summary document and expand it to have more detail in it, the thought was perhaps that could continue to evolve and grow directly into a real spec or if not that then grow enough specificity then the summary has done then we can re-code it into spec language.
- AP: I agree, for AAI/OIDC flows that is a relatively straightforward document to spec language. I don't think we have disused the format of the visas enough. Not been talked about enough. Craig proposed a structure and we have gone and said this is vaguely what it would look like. How to present a controlled access data with nested conditions in plain text?
- TC: excellent point, we need use case specifics that are not yet handled.

- AP: are we the passport visa group or the passport security group?
- MB: probably makes sense to have a fleshed out format before bringing it back. Helpful for directing the conversation. Nesting is going to be very hard, how can we fully replicate that and keep the whole token under 4k. Does anyone have the context for the need for conditions? Where that came from? How often they are being used? This would be useful to have, would this be a showstopper?
- ML: history of the conditions, typical for data owners that data access agreement in the context of the applications affiliation, if researcher leaves then data access permissions need to be revoked. At the time when researcher logs in, their first affiliation first retrieved and check they have continuing affiliation with that organization and that is coupled to the permission.
- MB: so this is an important use case?
- ML: it is a use case before passports, that it didn't really cover.
- TC: is it in the use case document?
- ML: I can easily add it there if you want.
- TC: if there are edge cases in the old spec then we need to consider those.
- MB: a simplification we can consider is limiting the level of depth in some way that it fits this common use case.
- AP: but eventually going to revient JSON these strings.
- ML: what does the word nested mean?
- AP: just two levels of structure, not a flat list, easy to represent in an array, minute you have a visa within another array, i saw those are conditions. That extra levels needs to represented somehow in a plain text string.
- So now you've reinvented JSON?
- MB: sounds right to me
- ML: not sure if a hierarchy but the structure to introduce to the conditions was quite powerful and rich in expression powers so maybe something if want to get rid of extra complexity we could simplify it a bit and keep the use cases.

Actions going forward

- Add to the migration story based on backwards compatibility between v1.0 and v1.2
- Continue on with the Summary v1.2 defining the details of v1.2 proposals
- Next week we will discuss the plain text details

2021-10-21: Token Endpoint, Architectural decision register and Version discussion

Chair: Alice Mann
Attendees - Name (Affiliation): Alice Mann (GA4GH), Kurt Rodarmer (NIH), Martin Kuba, Mikael Linden (ELIXIR-FI), Max Barkley (DNAstack), Andrew Patterson (UMCCR), David Bernick (Broad)

Apologies: Susan Fairely (GA4GH), Tom Conner (Broad)

	Actions Arising	Assigned To	Deadline
1	Submit examples of real visas to martin to compare with ELIXIR		
2	Migration story		
3	Email Andrew if you want to flesh out details for the architectural decision register		
4			
5.			
6.			

	Agenda Item	Person/Time
1.	Meeting time & feature table	
2.	Last meeting's Actions review	
3.	General check in on what next steps are	
4.		

Minutes

Token Endpoint decision

- David Bernick: Token Endpoint - using the official token exchange format to get the token, we thought verbally agreed upon, is that true?
- In the v1.2 spec we are putting in that getting the 4k passport token through a token endpoint is going to use that RFC or whatever as specified by max to get it. It does maintain OA, access token against broker to get 4k broker. Traced backable tokens and not an intermediary token. To me that preserves all the things we are talking about
- Is there consensus to using RFC token exchange

- Client interface - so not concerned to get consensus on that as broker want to use their own that;s ok. But if we say all brokers use this mechanism to generate their tokens
- KR: already been through the OIDC access token that has been returned, apprac token end point this time to retrieve our custom token which is a passport. 100% OIDC up until passport token comes back
- DB: when write v1.2 we write against that RFC, we don't need to invent a new way to get a token
- AP: this token exchange (TE) after OIDC flow?
- MK: as an implementer of authorisation server, it would be complicated to implement this RFC if the software used is not supported already. Would have to hack the software to how it works. As a standard from Jan 2020, good to follow a standard. Alternative use /userinfo to add new scope and claim to release a token as a string, also possible.
- DB: Not deprecating /userinfo stuff that already exists.
- MB: something to keep in mind, RFC for token exchange not a very heavy thing to implement, one request to make. Not a big deal to use for this use case vs other end point. As an implementer not significantly burden to do token exchange end point versus /userinfo end point with extra parameter. Similar amount of work. TE endpoint benefit if wanted to add more parameters in future, it has a framework for htat.
- TE: can specify some parameters for getting a token, /userinfo has no parameters. I
- AP: it naturally extends to downscoping etc. a good starting point on that journey
- DB: implementing where hack is difficult, what is the backend? Is it a standard thing?
- MK: miter
- AP: exchange mechanism not extendable in any way?
- MK: not very clear how they are put together. Not sure if it is a problem now, I already know how to hack for /userinfo and not sure if it has changed for token exchange.
- DB: **we really need to think about real world implementation. Knowing what tools and suites people use.**
- AP: if it was a complete custom end point is that reasonably easy to put in?
- DB: not deprecating /userinfo, can continue to support that. Other people might not. That still speaks to good interoperability.
- MB: do next part in your own code, not a lot of libraries that support token exchange. Hopefully that changes over time

Architectural decision register proposal

- AP: next steps, https://ga4gh.github.io/data-security/1.2-draft-add-adr/aai-adr
- Andy Yates' talk from LSG, maintaining an architectural decision register for specs. This seemed like an ideal way. Add a bit in spec where we do a non-obvious thing.
- **Email Andrew if you want to flesh it out on Original Authority.**
- Kurt: do a sort of v1.0 decision in the spec
- AM: Going forward we should make sure we are clear about decisions so we can record them.

Version Discussion

- MK: every visa issuer would need to implement a new way as to how to release visas for them
- KR: 4k has to be a v2.0
- AP: v1.2 could surely be a specification of token exchange, clarification of the flow, extra diagrams and better language but not actually introducing visas.
- V2.0 compact visas, has it been discussed enough in the Passport visa group?
- MB: the compact visa is the part that people want the most. Everything else is just in service to that.
- AP: I agree, not saying this in any attempt to not do the compact visas.
- MB: what if we focus on getting a compact visa format with token exchange, can make it backwards compatible if it hits the /userinfo endpoint.
- AP: only problem there is now, we have all visa producers. If broker has to provide a passport with either. Need to provide dual visas. Can we get all the visa producers to do?

- Cannot implement a client for both ELIXIR and NIH.
- AP: I totally support compact passports, I wanted a clarification as a client for how I go to two brokers, one in US or one in European. I need to programme my client as European or US at the moment.
- You can get some visas into a passport in the current format, you can't get many in there.
- MB: I think I agree
- DB: no good opinion on how to version things.
- /userinfo can still produce a fat token, token endpoint is different, getting a different kind of token.
- Who is it different for? For brokers an additive change, a new thing they need to do. For clients and resource servers: additive change for a client for RS changes what they would look at in a token. For visa issuers is the biggest change, we didn't discuss before. They have to generate visas in a different way.
- Brokers are usually visa issuers
- But above makes sense if they are different.
- KR: are we ruling out mixing visas of different types in the same passport.
- AP: we can think about that.
- AP: if i was expecting an array of encoded embedded visas, if i got embedded visas and occasional …..
- AM: take this to Susan and Jeremy to get an idea of if it needs to be a new spec.
- MB: we need a draft
- MK: I would like to experiment a bit with different ways on how to decode the token. Would like to ask if can submit examples of real visas so that i can compare how big they are. I have some visas by ELIXIRs they are mostly made of URLS and URIs that tend to be long. You probably wouldn't get much smaller by changing strings around these long things.

- KR: you don't need URLS as you are not supposed to follow them anyway. They are a convenient way to specify an identifier that has its own namespace. It has the same properties of uniqueness, doesn't require as many characters
- MK: you need a service that does the translation
- KR: yes but URLs in tokens are a security risk.
- MK: they are mostly used as URIs, unique but should be long so globally unique.
- KR: needs to be defanged, no venom. We are putting sharped knives into the drawer.
- MK: We need to support federations of 1000s of organisations. Manage some translation of identifiers from organisations
- KR: make them non-followable, introduce a useless scheme by doing that. It's trivial to introduce
- MK:
- KR: if you don't you aren't practicing safe security
- AP: are already global URI namespaces that aren't http. E.g. urn:fdc:agha.org.au:2018:id
- KR: fine to have some unique but also used to connect to server under token exchanger control, then dangerous
- MB: this seems like an implementer detail. We say use URI.
- AP: if you look at FHIR spec, URIs in form http is used throughout the entire spec. Entire field or health informations use URIs. they just write in spec, these are not things to follow these are just strings.
- KR: with a single user ID, security researcher could access 4 million records that did not belong with her. FHIR a good example of a bad practice
- AP: i also read the security report about FHIR, there are a few terrible implementations. Not a problem with actual FHIR spec
- KR: only saying that holding up existing examples and saying good example as it exists. Popularity doesn't translate to good practice
- MK: if you want to create for something for as many people as possible that is good
- KR: our task has nothing to do with genomics or any of the related medical fields or even science. Our task in this group, is to establish a secure foundation that rest of groups can build upon without worrying that what they are doing run into security risk. Our task is to create a foundation for other teams to use.
- DB: our job is somewhere in the middle. Highlight were there are risks and they are accepting their risk, saying where they aren't perfect. What we have landed at so far is to do something that is "good enough" and tooling exists for and we can get started.
- GA4GH is the genomics alliance. What we deal with is moderate security. What we do now is totally acceptable for that, but not for higher level security. If we are going to support what other governments support high.
- Multi stage plan. Plan 1 is get something that people can implement and heads them on the road to get good stuff.
- KR: i never said high, just not the lowest.
- If talking about genomics, focused on browser based. We haven't even gotten to the command line, how passports can support command line tools and in this case

passports are hidden disc. If these things leave their TLS tunnels, all bets are off. Usable by command line tools, our effort is not contemplating.
- DB: why not do both and think about a tighter future.
- MK: OAuth defines browsers and native clients (command line or applications on mobile phones). There is a definition on how to treat tokens in this way
- MK: our goal is to produce a specification to get globally adopted.
- DB: at the same time, just because more adoptable to do it the way we do it doesn't mean it;'s the right way to do it.
- For now it's the right direction. But there are things we can look at about the future.
- Just because adoptable doesn't mean it's good
- MB: passports were a success if only used for moderate data. But federated. Maybe can't get international places to share sensitive data. If everyone can share moderate sensitivity data if they were so inclined that would be a success.
- DB: data classifications from government, current standards are acceptable to every .. we deal with. Doesn't mean they aren't flawed, it's deemed an acceptable risk to use these standards. Good enough but doesn't mean they are always good.
- What is the way Europe or Australia views this?
- MB: compact passport format it may lead to v2.0, takes it longer to become a reality. Is it worth considering if other technical formats to achieve the same goals if use a more similar visa format.
- DB: v1.2 land token endpoint and compact passports. Then deal with the ramifications of that. Unless we deal with visa issuers who aren't brokers.
- AP: need a migration story otherwise don't end up with a better world. Otherwise every person implements two versions of visa.
- MB: two different kinds of keys. Want at least a transition period.use current format and change the keys?
- AP: what is the migration story
- KR: we are using a mixture of OAuth and non-OAuth. Passport token is not universally OAuth.
- *Discuss that next?*
- We need a decision and rationale for decisions taken in v1.0.

Decisions made this meeting:
- Token endpoint

2021-10-07:

Chair: Alice Mann
Attendees - Name (Affiliation): Alice Mann (GA4GH), Kurt Rodarmer (NIH), Martin Kuba, Mikael Linden (ELIXIR-FI), Max Barkley (DNAstack), Tom Conner, Heidi Sofia, Andrew Patterson (UMCCR) - joined 10 minutes late Susan Fairley (GA4GH)

	Actions Arising	Assigned To	Deadline

1	Send code to Martin to put in repo	Andrew Patterson	Done but only an hour before the meeting so may not have been merged by Martin (all my fault for leaving so late)
2	Will start the spec update	Kurt	
3			
4			
5.			
6.			

	Agenda Item	Person/Time
1.	Passports co-lead update	Alice
2.	October Connect Passports Workshop Agenda near finalised: https://docs.google.com/document/d/1o8o9jBpAok9MJYx53KjsYry7dJb2p4SVxTTa6hzGnXI/edit#	
3.	Last meeting's Actions review	
4.	Passport Size Discussion	Martin Kuba

Minutes

Notes from publication/Jekyll/Github actions work:

- Jekyll publish is working via github actions, with the plantuml extension to markdown included
- Publishing *only* occurs on branches named with the prefix `1.2-draft.`.
- The main working draft is published to https://ga4gh.github.io/data-security/1.2-draft-main/
- Each branch gets published to https://ga4gh.github.io/data-security/1.2-draft<rest of branch name> (for reasons known only to github, the trailing slash is required)
- PRs generated from forks cannot trigger github actions easily - so work needs to be done on branches in the actual data-security repo
- Branches should be branched off 1.2-draft-main (consider this the master branch for all the 1.2 drafting work) and should be self contained units of text if possible - a PR can

then be raised and the changes discussed, and then accepted into the 1.2-draft-main branch

Martin Size Discussion

- Tried an experiment where the visas can be made smaller by using different formats. Nearly same instead of plain text using binary representation
- Used in EU COVID pass, digitally assigned data into the smallest space possible
- Took a visa and measured how big it is, the size if 904 characters, exchanged signature with ECC signature, 3 types, smallest one went down to 648 characters
- Result is similar for two signatures
- Binary encoding with ECC key which produces binary data, binary data needs to be encoded into text, use base64 to use least possible number of bits. 512 characters and tried to compress it, 472 characters.
- We can achieve the visa to be small, 50% not something like 10%, even if we have 0.5 kb for one visa can put only 8 visas into 4kb.
- My question: is this the way to do it?
- KR: no, the proposal around making tokens smaller involves restructuring the entire claims approach and restructuring the visa approach, not just a matter of packing smaller. Curious why compressed base64
- DB: consensus on 4k passports look like, this stickers with JWT embedded token approach. We agreed 4k passports approach is different. I like this approach if we stick with embedded. V1.2 would be doing away with embedded passport.
- MB: couldn't this be combined? 4k passport has a combination of using the ECC algorithm and a change of what's in the visas to reduce them. Curious what size you get if you apply both these techniques?
- TC: don;t think the techniques fight with each other either. The work that has been done can be done and then encoding even in a more efficient way.
- KR: Emphasis the v1.2 that we started, did have embedded visas, they had a different representation, not represented as JWTs. is that you what you meant david?
- DB: yes
- AP: the discussed approach of using more of a plain text string and ECC encryption stuff, i did a check, the one i generated is 240 characters (not with affiliation and role etc).
- The newer ones are half the size you got to there
- KR: v1.2 moved to EC for signatures, no longer use the very verbose descriptions for the text fields if you could represent them in some other way. We just need to agree on something compact we can all understand. The treatment around the visas are still a thing. Some room for discussion around how visas in a string would look. I, for example, am not a fan of trying to put structured data into a string, i think structured data since it requires post parsing exposes a vulnerability or a potential attack surface. I would prefer to be careful about that
- AP: for instance the plain text string one I tried is the simplest of all the things. No where near as expressive as structured embedded visas, if I want to add an extra field to that controlled dataset here, I would have to invent some notation with some dots and now I

am embedding things that are parsed out of this string. This being a very simple notation here, only works for very simple cases and we might be better to stick with a structured JSON-y thing and as Martin suggested to do a more compact JWT notation of that.

- KR: can define canonical representations of JSONs, avoid the differences between JSOn representations that are equivalent as have unordered members and arbitrary white spaces etc. define canonical representations of JSONs that make that problem go away or get consistent processing. Those are things to consider.
- MB: do you have the code you have to generate these that people could try out.
- Martin's code: https://github.com/martin-kuba/ga4gh-compact-passports
- KR: concept of persistent passports - once you involve command line tools in the process, no longer in comfort zone of living within a browser and you get into some of the exposure of tokens on disc.
- DB: land on what the passport looks like first. Great topic for connect and we should talk about it. What are these passports going to look like? String that gets parsed? Is it now a recompressed JWTs (don't think it should be). I thought we were writing PRs now. So what is the consensus for what does the passport look like for v1.2. My opinion was not full JWTs but much more limited strings, fine if compressed even further but what is the representation so we can tell people at Connect. Don't feel like we have consensus on here.
- KR: spirit of putting visas in strings so that they could support something like a signature. But the arbitrary structure inside a string is what I was concerned about.
- DB: let's put that on paper and however we compress it whatever is best.
- AP: share his code with Martin
- AP: encryption test of that with no JS and a decryption test of that in python. Putting in string makes the signing easy. I do like the plain text of it, once decoded passport the thing we have in front of you is the claims. It is quite useful to be able to just decode this Passport JWT and see all claims. But worry that string not expressive enough, seems straightforward to represent this but more complexity you are reinventing JSON format in your own string format and that isn't good. My opinions on this current format.
- MB: hypothetically if we took the string format, kept the same content and gave everything JSON format and did binary encoding, would everyone be happy with that?
- AP: one compression advantage to the string approach, if you wanted to make an assertion of 15 different data access sets, put into one string and have a lot of signature. Some of compactness from here, not 8 different visas, each with their own signature. Can we introduce arrays into JSON structure, 10 different datasets into one visa assertion. Weren't going to require one whole structure for each assertion of every datasets. Could have 10 access datasets in the one visa assertion.
- MB: personal view is binary sounds interesting, would want to see the difference in size is, do that experiment, otherwise I couldn't say which one would be better. Have the same reservations about packing into a string, some benefits and some pitfalls.
- KR: one thing I'd like to remind people of, is JSON as a language is widely adopted but it's inappropriate for security purposes, no upper limit on string lengths or numeric lengths and any nesting, has a number of properties that make it difficult to support according to the JSON spec whilst easy to protect against attack. When using it for

Passports is that we don't really need unbounded nesting, don't need unbounded string lengths and might be well served while using JSON for expressing these, here are the maximum limits on some of these elements. If you then have a parsa that's so inclined. Easier to enforce from a security standpoint

- MK: format should be extensive, why JSON so widely adopted. Like XML but easier, can always put another field. Having a fixed… a problem, impossible to extend it if you need.
- TC: observe that the string shown that Andrew pasted to us, it's like JSON, property of extensibility if in future we wanted to add another claim. Agree with kurt but to not make it open ended and unlimited length and deep nesting, will all make it easier. If you think about parsa and what that looks like in code, general purpose will be more code and take longer to get done unless picking up existing libraries. This thing will small set of claims, easier to read and secure code review. I like the smallness and simplicity of this. I like idea of iterating as we go forward, compressing into binary rep, whether binary or string, essential data will be the same.
- Andrew:
  {
  "v": "c:8XZF4195109CIIERC35P577HAM et:1665130508 iu:https://nagim.dev/p/wjaha-ppqrg-10000 iv:39a277efae72236a",
  "k": "rfc8032-7.1-test1",
  "s": "FWAYv00igGtQVPv6GLmtzvh00LcysNXQGhLylXrS26whZxC_qZzS8t05a31gbisUFHHfDCX5inGSWi-IaUldDw"
  }
- MK: so you want to develop a new language for this?
- TC: I see the benefits of both arguments.
- DB: what andrew is proposing it's JSON not JWTs, i think we agreed that JSON is ok to use for this or no?
- AP: The canonical signed thing here is a set of claims in the visa, is itself a string that has the format and that itself could be JSON but just doing canonical representation of JSON in JSON hard but signing a string is easy.
- MK:
- KR: there's also protobufs which is well established and supported in every language. There are binary alternatives to JSON that can be used. The thing about whether there is a canonical representation of JSON used naked or put serialised JSON and put into a string. Probably roughly equivalent. Then we are back to having a language to describing the structure.
- MK: not easier to invent a new language.
- AP: major loss of size came about because if you do a compact binary rep then base64 encode it and then put that into passport and base64 encodes that. That is a source of a fair bit of loss of size. This was an attempt for inner claims not to be base64 encoded. Can we come up with a way to do digital signing of compact little JSON bits, that would solve things and not inventing a new format. But allowing us to provide signatures.
- DB: we understand that there will be overflow things and understood as a real Passport. We need to have something, this is part of stage 1 not being caught with having any

spec out in wrestling with these things. In version v1.0 we said it was going to be big enough so we allowed for the fat token to exist. It allows for the fat token we just don't encourage it.

- MK: spev v1.0, the token is small but you use the token to get the visas
- DB: yes you can use that passport for authentication. There are circumstances where even if we compress it could be big. We can't have a big passport, for uses in headers it has to be small.
- AP: on current large base64 encoded things, getting four visas into 1 4k token say, at 50% of that. I have some very limited cases where i hit four visas in a 4k, 20 would be better. Saving 25% or 50% really does add some value to the passport.
- KR: this why dbGAP, uses a single visa rather than separating each as a separate visa. Single visa with all permissions signed once. Doesn't fit doing operations on level of passport where you can shuffle the visa. Always have to send this to the originator to reedit
- TC: i think the representation whether it be binary or string representation we should be looking at the size of the code. Reuse of existing libraries that have already had a lot of scrutiny. Avoid the sprawling spec, this can help make this decision.
- DB: goal to put more info into them and put them into a header
- AP: limited number of visas before hit bearer tokens
- **MK: why do we want to put authorisation visas in headers?**
- **DB: circular conversion, answer is because, we had the idea we only going to handle fat passports and pass around via POST. pass in headers was normative and required and necessary. That's why we did all the work for 4k, option available to have meaningful visas available for headers. Hard sticking point and we have to make that possibility available.**
- **MK: you can put each visa into separate headers, can repeat the same header many times. If you want to just pass around visas split and put into separate visas. They are more like statements about the user, actually authorisations do you want to use something different.**
- **DB: key distinction as they carry the original authority of the original signer**
- **TC: it carries the signature wherever it goes**
- **DB: can we replace the OA token with a smaller traditional big token. Answer we came up with is no, there would be key DPs/adopters who wouldn't adopt it if that was the case. Clear that DP did not agree.**
- **DB: back to representative tokens that can be passed around, how do we make them small enough to use in authorisation headers? We came up with 4k headers and not JWTs.**
- DB: would it be useful to go down this road on the spec? Or do we need more discussion about what the 4k passports look like?
- KR: second this motion. Declare that for v1.2 it is written in JSON, using compact identifiers and our particular case from NIH we are going to define the structure of the visa in any event, up to us to define it and it will be a string. Visas or visas are embedded strings and we (NIH) will see how they are encoded and signed. We should thoroughly adopt EC signatures,

- TC: iterate if necessary about specifying the algorithm in signature. Spec currently doesn't specify? accommodate in the future.
- KR: original spec requires RSA. Kurt would also like us to formalise around key IDs as being required in the header. Be globally unique. NIH RAS team are not using globally unique identifiers and that is going to be problem.
- AP: why are unique IDs needed? Why a problem?
- KR: JWT spec is broken as it says you are intended to use JSON parsa to inspect part of token that has it yet to have its signature spec. Should never use JWT without first verifying signature. When we (NIH) process JWT first san header for verification keys that are known and trusted and use that to test key signature - if can't find well formed key ID then reject.
- MK:?
- KR: security vulnerability to put URL in there as people tempted to follow up and it's redundant. We say it has to have a key ID and recommend that it be globally unique if possible

AM: can I check all agree with DB's motion and OK to go ahead next week? Yesses noted. No further comments to the negative on this

- DB: Passports v1.2 will not be the end but will allow other parts of GA4GH with dependencies on Passports will be able to move forward. Nothing perfect and all a product of consensus. Just another way to do endpoints. Who will start writing it down?
- AM: Other news - David is now a co-lead. Passports and AAI need to be unified.
- KR: will take a stab at laying it out (AAI and Passports)
- DB: all will collaborate once there is something to collaborate on
- KR: does Andrew have anything want to start with?
- AP: OK with branching strategy off 1.2 main?
- KR: fine with that
- AP: will start some branches and then plenty of scope for an additional branch and people can review as separate PRs

Comment from Max:
FYI, I was modifying Martin's script in the background and encoding the string value as binary JSON + base64 makes it 2.5x the size (~100 bytes -> ~250 bytes)

2021-09-23: Spec Update and Audience Discussion

Chair: Alice Mann (GA4GH)
Attendees - Name (Affiliation): Max Barkley (DNAstack), Kurt Rodarmer (NIH), Martin Kuba, Mikael Linden, Tom Conner, Heidi Sofia, Andrew Patterson (UMCCR), David Bernick

	Actions Arising	Assigned To	Deadline

1	Get Jekyll set up	Andrew and Tom	Done
2	Define 4k token	Tbc	
3	PR for how you get a 4K token	Tbc	
4	PR for what is the additional security mechanism: aud or TLS for security replay attacks	Future discussion with all (post starting the work on 4k into spec)	
5.	Language clarification of terms	Andrew to set up a PR just with terms we need to clarify	
6.	Add NIH use case to the use case document	Kurt	Before October Connect (12th October)

	Agenda Item	Person/Time
1.	Apologies	Alice
2.	Passports co-lead update	Alice
3.	V1.2 Spec Update	Alice/All
4.	Picking back up from last week - audience vs TLS	All
5.		
6.		
7		

Minutes

Welcome to this week's meeting
Introduction of Martin Kuba as new co-lead for Passports, still to confirm third co-lead

Kurt summarising the group

- Focus initially was to create a single token mechanism for achieving a number of things
- Passports was initially rolled up in the DURI project, bone fidas initial use and other mechanisms to not pass authorisations properly but to pass evidence that might inspire an on the spot authorisation but inspire recipients. Allows access based on

good standing in some organisations, gave way fairly early on to more concrete authorisation, research owner. Tokens would carry either that information or access to it or your outlook on things.

- We have a token now concentrated on carrying specific authorisations, endorsed by the authoritative owner, assigned loosely to an authenticated entity or user.
- Main difference we have going on that this is not an identity based security system.
- Kurt worked on an identity based cheque for IBM back in 2000.
- Our pilot this was using a capability model for carrying the authorisations and the check values.
- The focus for going forward for v1.2 take what started as overly verbose expression of content of a token where people had expressed a preference for use or urls and use of urls in the tokens and use of human readable content even though no human was going to need to read them at high speed. This led to overly large tokens when all bundled together. Focus recently has turned towards a TES compact token still represents
- For v1.2 trying to ratify and yet all that that does is give us the same thing that we've had in v1.1 in a smaller version. We need to move toward basic token operations for subsetting the permission sets and also going to need to look at modifications for assignment to individuals and reducing the risk of bearer tokens.
- One of the things that people are confused about is there is a misconception that a single token can do all the operations and what a passport represents an authentication event or user and authorisations that that user has been granted by a resource owner or authority to access content. Nothing about what you have access to systems, not intended to give access to a system. That is kind of where we left it last time.
- What we've been pursuing in GA4GH is a base minimum interface between organisations, each organisations likely to have their own set of local requirements that may need to go beyond what can be put into GA4GH, Ga4Gh not about one organisations, so it establishes minimum standards
- Martin: portal for use interaction and storage elements and some computing elements that were processing the data.
- Kurt: KEYCOS had it in production in the 70s or 80s, this problem is well known already been solved. We've had a lot of challenge bringing people along on that. Expect mostly that everyone is cooperating in good faith, somebody who tries to steal data for example.
- Kurt: what we know is that the attempt to apply simple and popular sort of consumer app based techniques will not solve the problem. In order to solve the problem in keycos there were a number of assurances that could be applied in a secure operating system that we do not have in the environment we have today, number of techniques that can compensate for them, not implying with tokens we can get to that level but we can get to a reasonable level but we need to understand what
- Martin: problem not solved by OAuth or OIDC and does not solve the trust among resource servers.

- Kurt: yes that has been a large part of the discussion, spent about a year on this spinning our wheels, the problem that oauth solves and addresses is not the problem that we are trying to solve her. Problem that we are trying to address is different right from the get go, oauth resource owner in the oath model is not the user sitting behind the browser. Instead going to be an institution that has legal liabilities for dealing with the data
- Not overly difficult to apply a lot of security to the data but it is difficult to do that to enable widespread sharing to enable widespread research. Trying to maintain current security whilst increasing access, we don't want to decrease security to increase access. Looking at greater access without losing security and this means understanding the model. A lot of concentration on the maintenance of OA in the token flow that is being lost in the OAuth flow as the way it was extended.
- Martin: are we trying to write a spec for a known solution or are we trying to find a solution for something
- Kurt: it would be inappropriate in my view for a body like GA4GH that does not specialise in security to solve the general problem of tokens. That is what I work in but that may be a little over grandiose. In my opinion provide a solution that works for base use cases that GA4GH has, much like oauth not a global solution to everything, it is useful for a handful of use cases. For a tailored application of known techniques of two current use case.
- First efforts in passports did not represent the interests of the data holding institutions and they did not back the legal obligations and liabilities of the parties that were involved. We moved that slightly and slowly in a different direction that does represent those that can be used as a substitute for some of the legal processes are involved.
- This is as far as we got in a couple of years.

- AP: what I was hoping to get out of all of this. Content that is currently specified in passports spec is in practice unimplementable for a greenfield implementation as the spec documents two different ways of doing it, NIH way and European way of doing it. Hoping to get some kind of unification to some common way of doing things. Along the way lots of interesting things coming out.

- v1.1 (NIH) does not actually exist as a specification we can point people to, so you get v1.0 and v1.0 doesn't make clear the roles and responsibilities in the system. Clear from the discussion when I first joined that it wasn't clear to the implementers as to the roles and responsibilities. Not a good spec if you can implement two different ways that are incompatible with each other.

- Australia has no ELIXIR and no NIH but needs to work with both these brokers. We hope that along the way we solve that.

- Kurt: 1.1 is little more than an encapsulation of visas from 1.0, and we attempted to get it ratified in GA4GH. It has been specified, but died on the PR vine within GA4GH.

- KR: I'd also venture that when I signed off on v1.0, I did that an in the same breath I will sign it but we have to change it almost immediately. That v1.0 was unacceptable other than a stepping stone to v1.1. but for some reason that was never taken back to the specs. There isn't even a definition for passport in the v1.0 spec. strict and understandable definition for visa.

- Passport: access token plus claims

- Kurt: actually not. It was the intention. Passport described as a collection of visas. But no definition for what a passport looks like.

- Andrew's branch for v1.2: https://andrewpatto.github.io/data-security/
  (NOTE: proof of concept only to show how Jekyll publishing works - will move into real GA4GH branch - link/content may not be stable)

- In the repository.

- TC: improvement over just raw markdown and +1 Max. diagram looks really nice.

- AP: start this. I don't have any permissions into GA4GH repository, PRs against this v1.2 branch and people can comment on those.

- ML: discuss and agree on high level details before preparing the PR.

- ML: diagram useful, enough detail for discussion.

- AP: Tighten up the diagram to make it correct.

- Use cases share with Kurt:https://docs.google.com/document/d/1x9jEmSQqaFRDMrFAvCHtma0N0QrNfTKLG3xqpJRw8pE/edit#heading=h.vo7w4rw4ya22

- Let's get this 4K compact representation into a 1.2 draft ASAP. Getting these use cases in here.

- KR: I support this.

- MB: third degree is use case collecting.

· KR: Kurt needs compact passport soon as we need it for FHIR. Compact rep preferred to fat and skinny.

· Add NIH use document. And spec update for 4k

· Andrew's suggestion to make v1.2 PR branch right away. Introduce individual changes, agree as a group, what are the individual changes for considering. Just introducing 4K passports highest priority thing. Two things that could follow as separate items for discussions

    1. How do you get it, from initial log in
    2. Where does the audience fit in

- Could be done as separate discussions to avoid blocking things we all agree on due to more contentious things.

1. Get Jekyll set up
2. Define 4k token
3. PR for how do you get 4k token
4. PR for what is the additional security mech: AUD or TLS for replay attacks
5. Language clarification for terminology in anticipation, we can't call access token a passport access token - Andrew can do a PR to add a term and trying to clarify some of that language throughout.

- Kr: not put on table for passport spec, mechanism under consideration outside GA4GH that is being used, didn't imply everyone has to implement it. Is audience optional or not? Not able to be used or respected as design necessarily, we probably will not be using it. Same as OIDC scopes and scopes in general as these are not OAuth or OIDC tokens.
- MB: to Mikael's point for that particular point of audience I would be in favour of picking up the discussion.
- Other features discussed doing PRs for, these could be started as soon as we have the bandwidth to do so.
- TC: agree should be started. Start a threat model along lines of old OAuth threat model that lists potential attacks and defences against those for Passports and visas. To the extent that it is complete, audience vs TLS and/vs or both.
- Threat model not a standard, more written reflection on different types of threats available.
- Another tab on Jekyll, one spec, FAQs, threat model, use cases.
- In looking around in github repo on AAI found some fascinating pages of interesting discussions, fallen into repo land and unclear as part of the spec, added AAI about page as good intro. Better to paint a picture.
- TC: I'll get that started in parallel with other efforts max listed
- MB: how do we advance the first part of how to get the Jekyll page set up.

- We decided it would be post v1.2 to merge AAI and Passports spec.
- Start an email thread.

2021-09-16: Spec Update and Audience Discussion

Chair: Alice Mann (GA4GH)
Attendees - Name (Affiliation): Kurt Rodarmer (NIH), Max Barkley (DNAstack), Martin Kuba, Janne Lauros, Mikael Linden, Tom Conner, Heidi Sofia, Andrew Patterson (UMCCR)
Apologies: David Bernick

	Actions Arising	Assigned To	Deadline
1			
2			
3			
4			

	Agenda Item	Person/Time
1.	Apologies	Alice
2.	Passports Workshop Update	Alice
3.	Summary from last meeting i. Downscoping probably doesn't need to land in 1.2 ii. Open question about role of audience iii. Token Exchange - update from email conversation following last week's meeting	Alice/Max
4.	Clarify David's email > So from MY perspective it appears that the: > - client get jwt access token > - client sends access token to clearing house > - clearing house uses /token endpoint via token exchange to get 4k passport > - passport is used downstream > I BELIEVE that is what max is now proposing and it should maintain original authority (the real access token and the real passport are all that are used). I would term this a 'callback' use of the jwt access token (i.e. from clearing	Andrew

	house calling back to the broker) I thought Max was suggesting the 'client' use of the jwt access token	
5.		
6.		
7		

Minutes

Passports workshop

- Current snapshot on where we are with v1.2, by all means this isn't intended to be presented as a finalized piece of work but more a snapshot of the progress in technical discussions, outlining the main differences and changes.
- Is everyone comfortable with this?
- Driver project interest already: EGA, H3Africa, Cancer group and EUCANCan.
- Max: ASI? Contact them for the workshop

Andrew's question from the email thread:

- Send access token to clearinghouse and passport used downstream
- Client doing token exchange not the clearinghouse? is this a slip in the emails or have I got the wrong end of the stick.
- MB: also being slightly confused at that description, not what we discussed/diagram
- TC: did discuss the diagram in some depth.
- It was clear what the roles represented.
- MB: reached enough of a rough consensus
- Draft into the spec
- HS: NIH representative, I don't think there is a consensus, implications or consequences and trade offs with these decisions. Goal of this group is to figure out pragmatic paths forward, let's keep that door open and it will have direct consequences, this consensus or not consensus.
- MB: rough consensus, no one so opposed to what we discussed last meeting or in email thread, enough of interest to continue in that direction. Still a lot of details that need to be discussed.
- HS: hard to get everyone forward.

Summary from our discussions so far:

- KR: probably most positive development and seems to have wide consensus interest in compact form of passport. Happy about and not able to meet the implement, mid stream on the 1.1 version but we will be starting out a prototype of 1.2 fairly soon. Begin discussing with dbGAP on Friday. All good. Some of the projects that had the passport team had over the last year, downscoping mechanism and there's an

interrelationship with VRS and DRS not beneficial if we keep those linked, order to do things first, passports goes first as to here's how you can downscope. And make use of you to drive it. Got into a circular firing squad. Compact form I think that's good and can concentrate on wrapping that up fairly soon.

- Still only one of the issue we are looking at, understood or generally agreed to. Initial interaction purely OIDC and OAuth 2 leads to an access token that has a higher level of access semantics that we want to from passport and secondary level of token less powerful that is the passport, from email threads there seems to be general recognition that those two are not the same token- important. NIH places restriction places restriction that PIDC access token cannot be use as a passport, semantically different.

- MB: email thread Andrew referenced, relevant updates that happened between last week's meeting.

- It seems like we had reached a place where at least David felt like we were in a positive to try proposing a draft of v1.2. big enthusiasm for 4k token, the other points that were discussed in the email thread were the question of how audience plays into this. How the token exchange happened. A little of discussion on some of the details. Taking advantage of single sign on, maybe you don't need another exchange, user log in and just use cookies. Some kind of token exchanging is good

- ML: one of open question is role of downscoping, if we postpone or introduce here as well. Agree with analysis Max.
- MB: it's in enough of a position that we could try putting a draft together. David trying to get volunteers to write that.
- ML: corner cases and different flows. Can you request 4k passport in the beginning. If they serve the purpose.

- Author or review bandwidth.
- Ready for plenary to be reviewed.
- PR against this, text to change. Conversations in github can get unwieldy to comment.
- Do we need to amend Passports and AAI. Is David available?
- TC: Combination of the two would best be saved until after v1.2. so v1.2 can be more incremental update to existing spec, larger job of combining comes later. TC stepping in to be doing some more editing.
- AP: what's in a standard, a lot of value in there being explanation sections and sequence diagrams, not traditionally in a spec but I would put to the group that we should have some sections here. Some things in the spec that are 100% correct but not obvious. Not using initial OIDC token as a passport not explained in the current specification. This is non-normative but this is the rationale in the spec why this is this way.

- MB: plus one for me
- KR: question on whether part of spec but rationale past of spec
- AP: happy to type some of those into the PR
- MB: happy to do a draft. But can't start before plenary.
- AP: happy to start this.
- Access token, OIDC access token
- KR: it absolutely is the OIDC access token. We were calling it a passport token in the past, that was the issue, they aren't semantically the same though.
- KR: first stage of operation, up to OIDC and OAuth2, we start to diverge from that as that was GA4GH is trying to do, produce our own RFCs so other people can point that to GA4GH and say this is in a GA4GH spec. that's where

Max's diagram: https://drive.google.com/file/d/1yFyts--iPF12DFiis4khgFQKE-I-eiIe/view

Audience

- MB: open question as to what role audience will play.
- MB: will start from the requirements perspective. For ASI, to reiterate we have different Institutions, that want to release some data to researchers that are not necessarily affiliated with their own institutions but other Institutions. In practice this means the RS in an institution will receive a token hopefully a 4k passports with visas embedded in it. It might be looking for visas signed by one of the peer institutions, not necessarily itself.
- May be that Autism speaks is looking for if researcher affiliated with a number of other autism datasets.
- In that scenario, permission downscoping when we get there in future will be helpful, not necessarily as powerful as audience, may be many resource servers may be willing to give access based on the same visas, that you have access to some autism datasets. Don't want RS to have huge trust for each other so token for one RS at one time, if one RS becomes compromised. Want a token scoped for one RS so maintain loose level of trust in the network, hoping we can include aud in 1.2 spec so 4k token can have a particular audience.
- Any other comments on the use case? Or other requirements? Do any other institutes have this requirement?
- ML: what the audience is for. Who decides and how which audiences are included to passports, is it the client? Who requests the 4k passport from the broker, can client request an extensive list of audiences for passport or is there something that limits what the client can request? Is it the broker that decides eventual list of clients and based on what information?
- KR: your question is correct. The function of the token carrying authorisations is you can specifically cannot limit the destinations ***up front***, other mechanisms for doing that but building into the same tokens not helpful, particularly type of authorisations in passport are all about what data you can access, what content, not about where they live or what systems you are allowed to access. To address what systems you

can access, essentially what audience is trying to limit, that is a different mechanism. We've gone around and around about the actual security effect of an audience claim, voluntarily enforced. Heard it stated that audience prevents token from being used where it shouldn't. If the token arrives at the place is where there is a bad actor, the audience doesn't do anything about that. The bad actor would have to voluntarily say I'm not supposed to listen to this so I won't.

- AP: but the bad actor doesn't have the data. The good actor has the data
- MB: the point is that if they sent it to a good actor, the good actor wouldn't respect it
- The bad actor can send to a good actor and works fine as aud at good actor
- KR: the bad actor could send it to the good actor and that works just fine so you haven't achieve protecting the data.
- TC: bad actor can't specify audience as can't sign a token
- KR: they didn't need to. If they are trying to protect data and someone intercepts the token limited by aud, I can still get to the intended audience and get the data
- MB: we should clarify the threat model, maybe we are discussing different threat models here.
- KR: exactly what I am getting at, I hear a claim of what aud does that is beyond what it actually does. Yes it's true there are some scenarios that it can improve security, not all scenarios, it isn't an absolute thing. From our perspective at NIH, it won't hurt anything as long as considered an optional claim, it's does get you right back to Mikael's question though, how do you create a passport that can be sent out to work streams that are going to be spread across 1000 nodes in some cases. How to utilise how are we going to know where that is going to wind up and how that is needed. Not the purpose of the passport to control which RS you can access. Passport expressed the consent of resource owner for bearer to access content owned by the resource owner. Limiting to RS is typically going to be done by a different mechanism. But If this mechanism serves your purpose no harm to adding this. There would be a harm in saying it is a required member and it has to be specified then you get into the behaviour of what is going to happen if it's not there and used in a scenario where an audience doesn't come into play.

- MB: still not completely convinced we are discussing the same threat model. If ok to be there optionally that is something we can live with. Maybe it's more productive to assume that direction and leave the threat model discussion aside. Mikael's question to address: who decides what audiences are allowed?
- MB: to the first point as to what audiences are allowed, my suggestion be that should not be in spec, that should be an implementation detail, not allowed to address in token. We could make recommendations in the spec. In practice a system where a client can ask for any aud and broker just signs and unvalidated but can get tokens for arbitrary audience, would allow you an additional security control and be strictly more secure than without audience.
- AP: surely the default in absence of audience is a wildcard where you can use this anywhere. If the client says I want these limited set of audiences in token exchange

then the broker says I put those in the passport for you. No less secure that what is going to do in absence of audience as give you the passport for all audiences.

- TC: it allows the resource owner to still have rules about where that data can be used. Which audiences are going to be based on what's allowed.
- AP: client will have some information about which places the token will be used. There might be 3 RSs and that may be something that the client needs to deal with, ask for 3 audiences if going to 3 RS. This can be out of spec, a mechanism for the client to work out.
- HS: Is audience really an authentication function?
- ML: No, audience describes which resource server is allowed to consume the passport
- MB: when the user logs in first time and say which sources of data you would like the client to access. If model of usage already this client can send passport and anywhere in world willing to accept it with visas in it, you might as well let a client, request any audience and get tokens for it so that the client can try and compartmentalise the damage.
- MB: what are your thoughts Mikael on your question?
- ML: if audience is optional for RS to enforce, we don't have a security control there. If it is there needs to be respected by RS, if broker issues token without audience, powerful passport. which audiences are allowed? broker issues a passport without an audience, powerful passport, obviously can't be enforced in resource server. Is this something that a broker is supposed to manage and control. In broker do we have a small table and a list of audiences that client is allowed to request? if yes how do we manage that list? Is there some kind of out of band mechanism that the client uses to convince the broker?
- KR: is anyone taking into about TES or any other work streams attempting to decouple systems of execution and tools and all of that. Built in assumption to GA4GH ecosystem, we don't know where data lives and we don't where the tools will execute beforehand and audience expressly prevents that from working
- AP: a vision that you want to allow things to run anywhere isn't same as saying client doesn't know where it is sending this thing for or where it is accessing the data from. At some point the client will need to resolve this
- KR: work stream expressly doesn't, the work stream says have DRS ID going to ask a server, give back something to me entirely opaque, try to access data and if in place not foreseen back at the client when I logged in, then I'm dead.
- MB: when you go to access, assuming you have token with sufficient permissions to hit a DRS access end point, then you've gotten a signed URL and you're out of the realms of GA4GH.
- ML: have no audiences in passport at all and any API call must be authorised by the user using regular scopes in OAuth 2 tokens, don't need audiences at all.
- Jot them down and talk about the threat models and concern about multiple token approaches from different

- MAX: if we are considering other alternatives then might be good to jot them down and think about the threat levels. If I am a bad actor that has obtained a leaked, what stops me from logging into another system?
- KR: and max what you just expressed the basic misunderstanding of what passports serves, passport does not grant you access to a system. You cannot log in to a system using a passport, it's not about identity or who you are or gaining access to an account on a system. Audience is an acronymism from identity based security and tokens and in particularly passport tokens are not participants in that model. There are other tokens that you can use to continue to participate in that model but passports are not those. Passports designed to carry a resource owners' authorisation for the bearer to access content not systems. If using passport as gateway parameter to access a system that is a security flaw and that is a problem.
- TC: does the token not have a subject claim that specifies who/ gets presented as a bearer token but was the token was tied to an identify via a sub client
- KR: it is, for audit purposes. An NIH passport does contain a subject but the subject not identifiable in any way, it cannot be used to gain access to an account, used for logging purposes only. If something happens and there's access to data and want to trace that back, we have a route back to the broker and the entity that created the token to be able to recover the information but it's not being spread out. Identity based systems don't federate. You have to share identities and accounts across the federated systems. Whole idea of passports and premise of federation, based on having receiving system not have to know the identity for the researcher but instead recognise the identity of the authority behind authentication behind them to access data. There may be an established relationship between NIH etc, but system will not allow Kurt to log in under an account, can only access the thing that is expressed in the token. By access, retrieve the bits on my side of the connection, no way to access commands or do anything on the remote system, it is not applied on a passport.
- TC: it is implied you get the data from the system
- KR: right so there are a few concepts at play. Authorises you to access the bit patterns, that is independent of location and system. Another concept that does the housing system, resource server have the authority or authority to dispense and distribute the data to the calling system. In our case and NIH's case, when we want to have control over that mechanism, implying mutual TLS so that we have certificates on on both sides of the system so that one system can identify an another. Not using audience but using TLS that allows us to have an interaction operations to know the participants in that conversation. The distributing system knows where it is sending it's bits. Passport is not this mechanism, the attempt to overload passport to be a one token does it all, routed deeply in two concepts, identity based security, not what we are doing and passing one token with http header. We need to migrate to a system that understands this is the internet, multiple authorities that can't be moving into one token.
- AP: before when talking about GA4GH vision where compute can be anywhere, how does that fit in when need mutual TLS connections with anyone sharing data with.

- KR: it doesn't, I see GA4GH as trying to establish a very baseline system that allows different systems to interoperate. When you start to ratchet up security levels, you have to tie that down. There are ways to use tokens to buy you something intermediate to that. We may use API authorisation tokens to express the concept of one system talking to another. Not as strict as TLS certificates, ends up forming closed networks.
- Within a federation that is pre-ordained, that can still work. Try to have open world system is doesn't
- AP: others in the world what passports and visas attempt to do is to make a system that enables global sharing of genomic data
- KR: depends on how you classify genomic data, plant genomic data easily shared. Human data not so much. Confidential and sensitive data

2021-09-09:

Chair: Susan Fairley
Attendees - Name (Affiliation): Max Barkley, Andrew patterson, Tom Conner, Kurt Rodarmer, Mikael Linden
Apologies: Alice Mann (GA4GH)

	Actions Arising	Assigned To	Deadline
1			
2			
3			
4			

	Agenda Item	Person/Time
1.		
2.		
3.		
4.		

5.		
6.		
7		

Minutes

Max has updated the diagram. Review update, consider emailed questions from Mikeal Linden and if the current diagram works for others including NIH

Sharing the diagram and walk through from Max

Checking the assumptions Max shared by email

2. This diagram maintains original authority?
    a. Agreement, no objections raised
3. Fetching tokens v a self-contained token (self contained could be JWT like RAS or the 4k token). Max is thinking of these as categories of tokens and belives that this flow could accommodate both? Is that correct?
    a. Focus on where the visas are - self contained with visas inside as opposed to model where this is not the case.
    b. TC: seems a spec could accommodate both flows. Fetch user pointing to issuer is optional if have a self contained token. Understood correctly?
    c. MB: issuer part in diagram in MBs mind is an implementation detail. Coupled but not totally dependent. With 4k format might need to fetch visas everytime. In another form might fetch in a nightly batch.
    d. TC: misunderstanding, only issuer can sign visas
    e. MB: flow largely same for some cases but need to know which types are supported by the broker interacting with. Imagining some implementers may not implement all. On client side, if had a discovery endpoint to find out what supported, could then adapt request.
    f. KR: not NIH, 1.1 Have been harping on for years about making thinhs machine readable, leading to small tokens and 4k and NIH will be moving to those. NIH does not want large passports
    g. SF: clarification on 1.1, there isn't a 1.1 passport spec
    h. KR: NIH went ahead and tried to drive this through GA4GH but it was in conflict with the 1.0 spec which Craig described as intentionally loose.
    i. MB: apologies for describing the larger format as NIH
    j. KR: 1.1 (RAS) in flight. Will need to maintain for sme time and run both in parallel. 1.1 needed for dbGap work
    k. MB: is it correct that we just need to look at how to exchange a token for 4k format and not the 1.1 format.
    l. KR: people in NIH will need 1.1 tokens but shouldn't need to be looking inside them - clearing house is the main target for the tokens.

m. AP: if NIH going to 1.2 would be happy for NIH to deal with both types of passport?
n. KR: no diff at NIH clearing house for 1.1 and 1.2
o. AP: and NIH happy to take that burden on and have a clearing house that covers both?
p. KR: yes
q. MB: taking stock, for 1.2 flow don't need to consider getting a 1.1 token (covered ny the RAS /NIH spec). Those interacting with RAS would still interact with 1.1 but that doesn't need to be considered in this spec

4. Flow could be modified to handle self contained tokens, also mentioned in assumptions downscoping. Before would like to validate. Do we think talking about how scoping could work should be part of this? Or, simply describe the 4k flow spec in 1.2? Do we want to talk about downscoping in this?
    a. AP: don't think Australian group will get to this in short term. Happy with simple flow clarification and 4k but no current implementations
    b. TC: Broad should be able to accommodate a number of different things. If we have to do it we will, probably better if don't have it.
    c. KR: downscoping is removing permissions but also more. Only reason to have downscoping is to have different contexts for use and implications for identity based systems are huge. If trying to use these in such an identity ased environment, this will break. If SUsan has multiple passports context matters and the mapping of multiple passports to one identity doesn't work
    d. AP: don't see how this is the case? How would something key off identity
    e. KR: many systems trying set up a cheap bridge to object stores and only using tokens to populate their ACLs - those have one identity, leading to one set of permissions for an identity. If using tokens in that way will get surprising results. If using as passports designed to work, would need to keep passport until point of access. If taking passport through.
    f. AP: thanks for explanation Hadn't thought of doing the bad way so that hadn't occurred.
    g. MB: multiple things. Downscoping - first mechanical way - getting a subset. Another mechanical thing is juggling multiple tokens. That isn't covered here. Is that a correct second mechanism?
    h. KR: saw some early adopters was exactly this backward way. Move from token orientation to identity orientation. Use case here is that Susan may have two active passports at same time. Each token should have it's own behaviour without relying on just keying off the single Susan identity - as that falls aparts once there are concurrent "downscoped" (i.e. different) passports in play for Susan
    i. MB: what would be a win in a 1.2? Looking to passports being more integrated in more systems and potentially more types? What does that mean for 1.2? 1.2 may not need to consider all but should try to steer in correct direction.
    j. SF: noting shoddy note taking and clarifying if there has been misunderstand in previous spec would be good to update that in 1.2

k. MB: easier to get a spec if minimal. Do people have a current use case where they need to downscope? ASI no such need for now - OK with 4k
l. ML: current controls can use aud to control who can see a passport. Those controls help to limit the damage if a passport is leaked. Are there other controls in place? Trying to understand risks from leaks if there is a misbehaving client.
m. MB: in example included an audience parameter - an OAuth extension - could be done for any JWT type (4k or other). ASI want that security mechanism, open question on some AAI thread about if other people want this. MB would also like this. When thinking about downscoping had been thinking of reducing the number of visas (reducing power of token rather than where it can be used).
n. AP: reducing number of issuers would be an easy one. I have eight issuers but I could downscope, easy for broker to implement, no obvious downside for security. Maybe a simple broad brush mechanism?
o. KR: realise what just said sounds reasonable on surface. Should probably know DURI team working on downscoping for the last year and has details worked out. Somehow the discussion got derailed. What was targeted for 1.2 was intended to include downscoping - with indirect and compact expression. Tend to agree with Max that, right now, given the time period, best to leave alone. But downscoping not a mystery, a lot of work has already been done, including discussing end points. Lost a lot of time. DRS protocol updated to accept multiple passport tokens using POST to avoid overloading header, other reasons too. Multiple token issue is pertinent here. Can look at multiple tokens in light of passports but also other types of access that passports don't address (access to systems and APIs) and there will be DRS servers in the ecosystem accessing other tokens. Not the purpose of 4k to put this back into the header. If calling into a legacy system that doesn't understand then header useful but wouldn't want to design a new API that way.
p. AP: happy for this to be post 1.2
q. KR: had a security review and saw an issue with repackaging. Guidance for RAS is that can't repackage tokens. Not accepting any repackaged visas for the time being.
r. (David Bernick joins call)
s. SF: opinion from Mikeael?
t. ML: Agree with Kurt, can limit the damage from a token but not a strong immediate need in Elixir
u. KR: misconception that visa authorises a single thing, for many visas however it's an all or one case.
v. DB: had unserdtood that this was how it would work.
w. KR: had already been worrying about the size of tokens
x. DB: thinking about Broad nice to be able to get things for multiple places but that thst isn't reality is fine.
y. MB: getting close to end of time
z. MB: summary
    i. Downscoping probably doesn't need to land in 1.2

ii. Open question about role of audience - some projects want but mightnot be desired for NIH use case. MB suggesting this as a topic for a further call. DB: don't think this gates 1.2. MB: would be happy with the outcome where it can be done as need for ASI but can hold off for 1.2. ML: thinking when we have 1.2 could do a security risk assessment to identify pain points. For example, a rogue client? How could the list of audiences be limited? DB: volunteers Tom to build threat models

iii. This flow, broader question, token exchange - could we live with the flow in this diagram or are there things that would need to change?

5. Close

2021-09-02:

Chair: Susan Fairley
Attendees - Name (Affiliation): Max Barkley (DNAstack), Alice Mann (GA4GH), Tom Conner (Broad), Andrew Patterson (UMCCR), Mikael Linden (ELIXIR), Heidi Sofia (NHGRI), David Bernick (Broad)

	Actions Arising	Assigned To	Deadline
1	Max to add notes on schematic and to send this in email looking for NIH input on this current proposal	Max Barkley	
2			
3			
4			

	Agenda Item	Person/Time
1.		
2.		
3.		
4.		
5.		

6.		
7		

Minutes

Use cases so far here:
https://docs.google.com/document/d/1x9jEmSQqaFRDMrFAvCHtma0N0QrNfTKLG3xqpJRw8pE/edit#heading=h.vo7w4rw4ya22

Reviewing Max's updated schematic

- Updated diagram here
- May need downloading to view fully

Feedback

- ML: is it an access token or self contained token?
- DB: the vision I have that the access token different to self contained token in a the traditional OAuth way, but self contained token is an access token in a non-oauth way, perhaps we should stop calling it an access token because that is confusing.
- MB: this response format, one area where using OAuth, confirming with that standard has a cost conceptually. The token you return is in attribute is called access token just because of the OAuth.
- DB that's ok but for convo we should make sure this should be defined
- MB: in ASI we want to get back a token we can use as an access token but can have different token types for different use cases if we need them
- TC: great, it fills in some of the detail. What was in the token? New diagram clear. Visa issued by the broker, signs the visa. It didn't carry forward from someone else?
- AP: in Australian genomics model I am imagining using CI login as a broker service, we are going to have some back channel by which we talk to it and give it a stream of visas on demand. But the visas come from the data access committee somewhere else. They are signed by DA software. Broker is not resigning them, just putting them into the passport.
- TC: could we put a separate note to be able to follow the chain of trust so you can see who is signing what. Going back to the original authority question, we need to make sure it's clearly covered in the document even if it is a footnote.
- Susan: Kurt not on the call today, as soon as we talk about exchange of tokens we come back to the concept of original authority. Is there some sort of problem here? Based on diagram looking at how does everybody feel about this with maintaining original authority.
- DB: deal breaker for NIH but question is should we allow for this in the spec. accommodate for others, other places may not care about this concept. Should one force the other and should we allow different level of flows? We keep coming back to diametrically opposed fews, can we accommodate both?
- HS: agree if we can find the ways where it most gracefully accommodates multiple

approaches this is the best we can achieve. Do we need to think about this ahead of time? Are there such ways of interfacing these different security needs or are there mistakes that can be made not thinking about this ahead of time? What are the implications?

- DB: implications is that downstream DRS and other component servers need to know what is coming to them. It ends up being something that continues to cause some consternation. You need to know that the incoming request to you is full OAuth or custom. There aren't many organisations to totally implement something from scratch. We can at the Broad and some places don't and that will hurt interoperability.
- HS: so security and interoperability. How do we best accommodate those two things?
- SF: my understanding is that we get a security benefit for re-using widely used infrastructure. More people who adopt it and broader range of industries used across, better maintained it is. Small group building this - will be a substantial load on this.
- DB: pragmatically, if in a large organisation that needs any internet facing thing that goes through an audit, if using off the shelf and well regarded used, then fine but if using your own thing then you start getting a lot of scrutiny and Institutions may not use it especially for getting access to something. That is the reality. Hard for organisations to say yes to using a new thing.
- HS: well tested/vetted etc and ease of evaluating what you built I agree. As a counterpoint you could say that older standards are out of date now, maybe if the world is moving to zero trust, maybe they don't enable that. What was suitable decades ago is no longer suitable. I have no ability to know so just asking the question. Probably a reality that NIH is going to try to move forwards with stricter security or perceived to be appropriate security for the systems. What are the best choices to be made in that context? Old not necessarily better
- DB: not against having new things. If we are suggesting new things and on the hook for building real referenceable tools, we have to offer software solutions to new ways, open source project and everything that we sort out. Funded and built etc.
- TC: documentation communication makes it very clear why you can trust the design fundamentally is an absolute necessity. Showing where the signatures are for the visas, look at the diagram and follow in your mind, who am i trusting? Who has the authority, who is going to authorise things. Sequence diagram makes it clear subject to quality of the implementation. Working code or just an idea? Sequence diagram lets just demonstrate to NIH we can scrutinise.
- SF: what about NIH counterparts in other part of the world. Requirements probably similar to NIH. Nobody here is relaxed about security, security is a foundational workstream, everything else has to come after. Have we explored shared requirements properly? We want federation system to work and adoptable and re-use current infrastructure. How do we support those things that we need?
- ML: What is the difference between the 4k token and regular access token, is it really that much different what access tokens are supposed to be?
- MB: passport access token or in general? My interpretation you could use to use this self contained passport as a regular oauth 2 bearer token, small enough to fit in a header, it has all the normal oauth JWT security mechanisms, aud and issuer and it has visas

embedded in it so no need to make additional requests to validate those other than looking up signatures

- DB: main difference is, trade one, size problem earlier, could user info be big and access token no? May not have separation now. Other part, originally access token meant to be 1:1, client to a broker. User info used downstream in a more federated model. If no longer necessary then discuss use access token downstream itself at CHs. access token supposed to be used at one broker and derived user info to be used downstream at n number of places. That's the big differences. Does that make sense in a 4k world? Limitations may not present now. Access token considered a route token and a derived token slightly less powerful
- MB: calling out one thing, filled in 4k token to conceptualise a way you can use it. Most important part of diagram not token you get back, you can exchange one kind of passport token for another passport token. Truly you could ask for an issued type another passport token with different aud, could be v1.0 where you have to call user info to get a list of visas. That would make a difference for how RS behaved but wouldn't structurally change the relationship. Critical, hope we can encode audience important in the spec as that is useful for my use case. 4k hopefully broadly applicable use case of that.
- AP: just confirming that the token exchange has no semantics of consuming old tokens - like oauth mechanism where there is a flow that consumes refresh token and invalidate the old token by issuing the new one? None of this here? Want to be clear that we could using the original thing 5 times to get 5 different things
- MB: yes correct, token not consumed in exchange
- DB: i think having token exchange is great but non-starter for NIH. can we not have two ways? But this is problematic so where do we go from that, continual sticking point. We can't force a DP to use it if they don't want to
- SF: NIH it seems feel that token exchange is not applicable for them. Is it truly the case that NIH's requirements truly different from everyone elses? Can we accommodate both with a bridge to make it clear which system they are in. we need to make sure we are confident that it is two different sets of requirements catered for here
- DB: original token sacrosent unless used in internal loop system. Public facing DRS server, and hands back a passport, problem for NIH workflows. This is an idealistic position, not a pragmatic one. In modern cloud environment you are giving them an OAuth token and your identity not tied to that token. NIH says idealistically wrong and my stance is that is reality

Key point here where we need clarification from NIH:

- **MB: misunderstanding where we are using token exchange to explain it in different places. Just how you go from one kind of passport to another, could be passport downscoping endpoint. My point of confusion, is that I thought we had got a consensus but not sure if this should be an oauth end point or a brand new end point but we need something where can pass in passport v1.0 token and get back a self contained token or a downscoped token, still going back to broker which is going back to issuers which are the original authority.**

- DB: agree with doing things that way but not sure there is consensus on this. We need NIH to say yes this is the way we understand it too. Then we can move on!
- MB: in contrast, the approach i was advocating before, even after getting self contained passport not send directly to RS, go to RS authorisation server and do another exchange and Kurt has advocating against strongly so not here in diagram.
- TC: but that isn't prohibited by anything. But what if token exchange at one RS and not violating authority here, that's just an implementation choice private to RS, is this naive?
- MB: I think kurt was agreeing but objected that this client not in same org as RS should participate in additional TE where every Institute they go to they trade in their token that is for that one institute. So client has to understand this so now part of standard way of operating, but I think we decided to exclude that. Distinct to here as getting an exchange of one sort of passport for another.
- TC: has to be part of standard that gets wider agreement.
- DB: if you can, can you post a message to the mailing list summarising this to the mailing list and hear if any last minute objections. If NIH is cool with what is shown here, we can put into the spec
- SF: can we capture this is the diagram for others coming to the convo
- DB: if this lands with NIH then let's make this work.
- MB: yes I will take that action item
- SF: how does this model stack up to current Passports v1.0. What are consequences for current implementations
- MB: this diagram starts with v1.0 at the beginning. If you were following strictly v1.0 and not doing anything else that deviated from that. This box here is where the data is, this client would be the clearinghouse and know how to get a flow to get a passport and know how to get visas. Reality probably passing that to somewhere. Don't think it is incompatible but not sure who has implemented something that rigidly follows the v1.0 spec
- ML: we have an implementation that follows v1.0. This extends it. One of key motivations, ability to downscope the route passport. Now if I understand, it is now done by a token endpoint call which sends a master access token and receives another token in exchange. That is good. It still exposes the master passport to the client that then downscopes it. Even extend the first interactions between the client and the broker so that the client can directly request the downscoped passport, no need to expose the route passport to the client at all?
- AP: yes good but look at what the things are in token exchange. What can you specify in the token exchange and go into original AAI passport flow. Where do you get to put in extra requests other than oauth scopes.
- ML: done in initial call of authorisations.
- DB: spec does say you can have other scopes, you just need certain ones
- AP: great if downscoping mech matched with downscoping token exchange
- MB: You can use the resource indicators.
- TC: full passport what used you are entitled to downscoped passport. Mikael saying need some smaller proof of identity and entitlement
- MB: first flow when user logging in client, saying only give me visas x, y and z

- ML: yes doesn't exclude further downscoping further in the flow
- SF: sounds like we have made some progress. Max, a couple of things come from questions, maybe in some form adding note to doc where we see this happening in relation to colocation of client and RS and points flagged that were not happening here. Adding some note on Mikael made, the potential for having limited scope at entry point to this system. Conscious that this is a complicated and detailed discussion, can we capture this in one place.
- So add notes to this
- AP: What is the sequence diagram that NIH already has? What is the actual reality here. That top bit is de facto the same as oauth. What is the sequence to visa data in the NIH system currently
- DB: currently v1.1 stuff on the RAS site. It's its own token format, not one that is an official GA4GH thing. Terra and Uchicago, Gen3 building for as NIH supports. NIH says when this group reaches consensus then NIH will adopt this. V1.1 uses v1.0 a lot, access token from broker, go back to user info and gives back a JWT format and format is v1.0 spec. Regular old oauth, handshake with broker and get an access token and put access token back to broker. Old school, simple flow.
- DB: old one, if user info overloaded is a problem big sticking point. And output handled in various DRS- too much questioning of things. This is trying to get things into a stream of everyone using this. Old one can be used on too many different ways.
- TC: old one used on backchannel auth? Not in spec? And this makes it more transparent
- DB: main issue was user info endpoint became too overloaded, we put too much stuff in it
- MB: whether or not it fits in header is a practical concern and downstream servers
- DB: once user info end point became a token endpoint and token became giant.

2021-08-26:

Chair: Susan Fairley
Attendees - Name (Affiliation): Kurt Rodarmer (NIH), Max Barkley (DNAstack), Susan Fairley (GA4GH), Alice Mann (GA4GH), Tom Conner (Broad), Andrew Patterson (UMCCR), David Bernick (Broad), Mikael Linden (ELIXIR), Janne Lauros, Heidi Sofia (NHGRI)

	Actions Arising	Assigned To	Deadline
1	Capture discussion in aud discussion for actual use cases in a document and we can link through to that from meeting minutes This document is now made and ready for editing here:	Kurt/Max Alice to start the document for editing	

	https://docs.google.com/document/d/1x9jEmSQqaFRDMrFAvCHtma0N0QrNfTKLG3xqpJRw8pE/edit		
2	Send any passports implementers or anyone thinking of implementing to Alice (alice.mann@ga4gh.org)	All	
3	Max to add detail to his flow schematic on last point particularly Updated diagram here	Max	
4	Number 9 in design doc, needs updating with recent aud discussion	Max and Kurt	

	Agenda Item	Person/Time
1.	Frame of next steps and timeline	Craig
2.	Exchange token fit with requirements	Max/Kurt
3.	Discuss audience mechanism or alternative	All
4.		
5.		
6.		
7		

Minutes

- Andrew's example: https://didact-patto.dev.umccr.org/api/visatest
- New plain text visas signed with EdDSA as examples of visa length
- Should be stable for a while
- Showed that the compressed token is about a quarter of the size than previously.

- Susan: We had some email discussion on aud (see aai google group) and some useful discussion to specific use cases and the role of aud in different use cases. Should we review the current set of use cases to make sure this is captured?

- DB: it sounds like we had reasonable agreement on the aud discussion so can write this into spec.

- My only Q: does it only exist in access token or does it exist in the self contained passport itself? But other than that the agreement is that it is useful under certain circumstances.
- MB: sketched up a small sequence diagram of how we hope to use aud as part of the ASI. Out of that aai email thread, became clear that there are two categories of use cases for an NIH like scenario where resource servers have data that is owned in a sense by the visa issuers or a federated system like ASI where the visa issuers don't own the data.
- SF: are these situations adequately captured in the use case documents.
- Craig: Start a use case document? How many docs? Definitely in the summary doc we have. First half of doc understanding the set up, policy side of folks and a summary of the use cases for policy people to make sure solving the right problem. And the second half of the doc would be technical details and make this doc as short as possible.
- Also useful to have how the general case for people's needs
- KR: traditionally we elaborate specific use cases and try to generalise, general use case with eye to federation, our cases yesterday are going in the opposite direction than general federation. Lock down specific use cases and make tokens not generically applicable. Max identified a specific use case where we need that and at NIH we also have specific cases where we need to lock it down. It's the opposite direction to what GA4GH wants to get to which is federation.
- SF: useful to come back to specific real use cases, if we go for something that is general but not solving anyone's real problem so let's capture these specific things.
- Craig: a new number 9 to our table of issues, my historical understanding, so needs updating. Might need another summary section, overview of the strategy "do this and do that if you do this etc"

Agenda item 1: framing of next steps and timeline

- Craig: discussion is critical and we also went to get feedback from the broader audience.
- Understand our stakeholder's timelines: NIH needs to move ahead with implementation and ASI needs to move ahead with implementations.
- Bring up timelines we are aware of and we have goals for ourselves set up even the Plenary, where do we see ourselves being by the plenary and by the Connect meetings
- We have alignment in terms of our progress and what we wanted to achieve
- Work in progress for NIH implementation right now.
- Ecosystem around NIH and RAS initiative already has something in play, skipped over v1.1 to understand what makes sense for the overall community. These things have to keep moving forward inside NIH. NIH does want to move ahead with what they have now but will be a stepping stone to get there. The current NIH initiative will have large passports and we need support to move over to whatever we come up with (4K passports, smaller ones). We will need to support moving over and the switch to the 4K passports. We need to discuss this in terms of timelines. When will we develop v1.2 and NIH can then implement it and make sure it rolls out.
- How do we feel as a group as to what pace we want to move at?

- SF: do we feel that amongst this group we have key passports implementers? Can Mikael speak to EGA from an ELIXIR standpoint? Do we need to get in touch with other key passports implementers
- Craig: the doc that we want to prepare would be part of the outreach, we need a deadline for that too.
- Mikael: EGA integrates via ELIXIR, OIDC provider. Can't see any dramatic implications to the EGA integration. We can bring them to these calls as well. Mikael has contacts there.
- SF: if Mikael is happy to give a broader overview of their position then that's great.
- Kurt: I thought that anyone who had been sitting through my ramblings would have gotten the idea that NIH was really putting a lot of muscle into this effort. But in case that's been missed, for several months we've had a stable and standardised representation of the GA4GH passport, could not be called v1.0 as made into a token whereas v1.0 was undefined. We called it v1.1, it is a self contained passport using GA4GH requirements of how to define things with URLs and wasteful with info to bits or bits to info ratios. This is what led us to look at 4k compact representation and Andrew's experiments sounded very encouraging.
- We've had this going for some time and not just NIH but all of NIH partner systems are also putting significant engineering effort into this. We have several partner systems that are working on this. The ability for us to drop that and move to something different is zero. We're in flight right now and we have run through 2 or 3 implementations of RAS. Becoming an NIH wide standard, it is taking route and not likely to be looking back. Anxious to bring in a compact representation as soon as possible as we have reasons that the non-compact version will not address. That said I have a handful of access privileges more than an average researcher and my passport is right now about 6500 bytes, fat RAS self-contained passport. Can get larger than that, but when we talk about these passports being large and being under 4K, it's likely that many researchers even with fat passports will be under 4k. But we can't count on that we can't build a system that fails 40% of the time due to the size. But expect that majority of time will be small enough to fit under the 4K limit. It's not just in theory, it's something we are doing and have been doing for the last couple of years, well in-grained and not much chance of reversal of course, no chance really due to amount of money spent and we've got a lot of people working in this direction.
- https://auth.nih.gov/docs/RAS/serviceofferings.html
- v1.1 format in the middle of page
- DB: mostly aligned with GA4GH spec that has been written but diverge with v1.1
- SF: current discussions of passports on v1.2, Kurt's is v1.1
- DB: we said we would skip v1.1 so not to be confusing and unofficially say NIH's v1.1 is v1.1 and we are discussing v1.2
- Summary of path forward: https://docs.google.com/document/d/1lSRIJRFSIB8EMww_yOY6hWkT6O7jDABdmxegsmFBD24/edit#bookmark=id.oq3uvc38lg7w
- Kurt: it is quite possible that we will see v1.1 fade away, not initially because of the slow moving nature of development. We will introduce v1.2 in parallel with v1.1, they will both

exist and only those systems that are reaching inside the tokens and trying to deal with them directly and personally, those are the only systems that would care. The vast majority of system using passports or use them as gateways to access data would never see the difference, in terms of whether they post or use these things in headers. Definitely would not recommend using headers even if small, but this can't be avoided due to backward compatibility. So one reason why we need to introduce v1.2 is that we can't work around that any other way, the affected parties are passport issuers and the clearinghouses.

- DB: for NIH it's broad, UChicago, UCSC
- Craig: and SRA through NCBI / dbGAP
- DB: format might be different but fundamentals still the same, not a total rip out and redo. Implementing Passports period is a big change but the change between versions is not that overwhelming.
- Kurt: treat tokens as opaque, then that leaves only a few entities looking inside and doing something with them. If you are using a clearinghouse rather than doing your own processing, then it's the clearinghouse that gets updated and carries on and clients with no impact
- DB: speaking as a clearinghouse owner there is work to be done on this.
- Craig: in the interest of time, do we feel we are at a time where we can wrap up most of this
- **Capture if anyone does have timelines: we need this useful summary document for Plenary and share in advance of October Connect**
- **Summary document to be ready for Plenary/October Connect**
- **Have a strategy, get feedback and move forward with some of this.**
- **Susan to take ownership for creating that document**

Agenda Item 2: Exchange token fit with requirements

- Craig: still an area to be decided upon
- Max and Kurt tend to discuss the two columns on the design doc. It's not well understood what we want to going forward
- MB: schematic and trying to bring this into the aai mailing list discussion. Two sets of requirements for two different use cases between NIH and ASI
- Showing the flow of tokens for the ASI use case.
- In ASI participating organisations that have data on people with autism. Share data with researchers that are associated with other organisations based on other researchers having been vetted by others. "That's good enough for us so you can access some of our data too"
- It will be up to each institution to decide their own policies and what level of sharing, it's a loosely affiliated network, minimal amount of trust between individual institutions, will have to have a single broker of passports and visas that they will trust to report these things accurately.
- Showing how in ASI that a researcher user will log in through a clearinghouse and a broker and get a passport and do one exchange to get a new potentially self contained

passport that has an audience at one end. Passport flow through version v1.0, interactive log in, now clearinghouse has a passport access token for them. This point they haven;'t committed to a single dataset yet, have a passport access token that could take any visas.

- Now send a request to clearinghouse for data, a single exchange of tokens, key ingredient that there is some resource server, that is audience of token. Get a token back will only be useable at that single resource server. Token like a self contained passport, GA4GH would define here. If end point that isn't oauth or make a custom end point, key ingredient say the audience of where we may use this and in terms of what RS I plan to use.
- If the request works, get back self contained passport and go to use it at your RS.
- That is in essence the core flow for ASI so we can get tokens that aren't just downscoped by visas you have but where can they be used.
- Downscoping based on what visas and downscoping based on audience are the two key ingredients to this end point, will make it work for a lot of use cases not just ASI
- DB: is there a way that we can programmatically differentiate between those doing token exchange and original authority passports, these are two opposites here?
- MB: no, there are places where you can do token exchange where different semantics but here exchange with same broker that you got the same passport originally. Analytics environment is the clearinghouse for example. Always go back to the same broker everytime they want to access a new RS. visas signed by the visa issuers. Does that clear it up?
- Andrew: host is RS, but host should be broker?
- (yes human error above)
- MB: host should be broker, audience is the RS.
- Andrew: ability to specify multiple audiences?
- MB: might be use cases for that so not disallow it but might want to encourage people to pick as few audiences at once for security considerations.
- AP: more like passing it to a downstream compute service who will then pass to downstream service. Can you make sure it can be used by all of them? I like using the token exchange endpoint, feels like what the token exchange spec is meant for.
- Craig: as for resource indicators could possibly put the visa issuer url as the start and then a space deliminated set of compact visas that you want included for that issuer.
- TC: seems like tokens are short lived tokens, narrower in scope? Access to fewer things than original passport. Are they limited in terms of time?
- MB: yes there are cases of lingering long lived work flow issue that always comes up when you talk about token exchanges.
- TC: refresh same token for some period of time?
- MB: we could have a refresh token. Go back to the source periodically
- TC: clearinghouse is where the analysis is happening so this would work fine in this use case as you describe it.
- MB: CH are workflow analysis places. May also be third party services. Brokers for ASI would only be a single broker. This could be relevant to other consortia in the future, a broker for your network and RS individual services amongst the participating institutions

- DB: is the user agent sending the access token to the CH in this model and that sends onto broker?
- MB: here CH some sort of web app, user agent may just get a cookie that binds it to a web session of some kind
- Kurt: what support for command line tools? 100% of genomic processing happens as command line tools
- MB: here web app with jupyter notebook in it but could rejib where CH is a command line tool and thus also user agent and collapse some of these things. Another option CLI goes through CH and it has some other proprietary mechanism.
- We haven't fully narrowed it down in ASI use case yet
- Kurt: i see the user who is represented by a user agent netiehr have any authority in the flow. I see a CH and a broker and RS that are supposedly in a position of obeying the instructions they receive in the tokens and the requests that bear the tokens, no representation of the data owners or authoritative responsible parties?
- MB: yes not included those in this diagram, broker when issues tokens has some connection to visa issuers. How things are packaged fine but how broker gets them are not defined.
- RS has some component in them not defined by standard but instructs them as to what visas they are looking for in what tokens they receive.
- Mechanisms will exist in ASI but maybe don't fit into what needs to be standardised.
- DB: doesn\t breaK original authority as CH gets self contained token from the broker
- Kurt: no, not correct. The placement of CH in flow is critical to preservation of OA. if the authority issues token and within a few milliseconds go to a CH and say is this token valid and exchange for something else and then have rest of flow without inclusion of original token or authority behind it, yes eliminated it from entire flow. Proper placement of a CH or any entity that has the function of validating OA is at the RS. all the way at the end of the flow.
- MB: last request using some kind of passport token, meant to be perhaps the 4K token format or perhaps the v1.1 NIH self contained format
- Kurt: that is irrelevant.
- MB: that is a token that is used by the broker and contains visas signed by the visa issuers. So how has it lost the OA as that is what the RS will see?
- MB: CH has a technical meaning in passports spec that is the meaning I intend her. Could mean a service that inspects a passport, that is not how i am using it. Here CH defined as v1.0 spec. RS receives a token signed by broker and each visa signed by the issuer of that visa. Work backwards, start at top and do a flow and get an access token that is signed by the broker and exchange it with signed visas but still have one contained thing that is signed correctly. Where we try to do it, we do one exchange, here is my access token signed by broker, please give me back a self contained token both signed by you and contains visas signed by the visa issuers. RS relationship of trust with CH is minimal. There is always a risk that maybe CH is not the service I think it is and token leaked somehow, always a concern with bearer tokens. It can look into that token it receives itself. It has visas signed by the visa issuers I trust.

2021-08-12: Exploring proposed 4k passports

Chair: Susan Fairley
Attendees - Name (Affiliation): Kurt Rodarmer (NIH), Max Barkley (DNAstack), Susan Fairley (GA4GH), Alice Mann (GA4GH), Tom Conner (Broad), Andrew Patterson (UMCCR), David Bernick (Broad), Mikael Linden (ELIXIR)

	Actions Arising	Assigned To	Deadline
1	Capture the summary points in the design doc	Craig	
2			
3			
4			

	Agenda Item	Person/Time
1.		
2.		
3.		

Summary of our discussion today

Useful to capture the follow things in the design doc:

a) **Discussion of use of user info**
b) **Maintaining the whole passport**
c) **Downscoping**
d) **Audience mechanisms**

Introduction

- **We ended by getting agreement on exploring 4K passports. We had some initial exploration and reached the stage where there was one of two questions on the detail, on Mikael and Andrew.**
- **We will start with continuing the exploration of 4k passports.**

- Andrew: **I can cross off one of my concerns**. I had a play with cryptography libraries and asked about at UMCCR. General vibe not using a JWT library still ample opportunity to get access in python and same encryption stuff that JWT use,

not too hard to unbundle it from JWT. Think it would require non-JWT visas. Require us to write something extra and keep in github with some standard examples.

- David: if we did it in python and two other languages and sub library to the existing JWT stuff and made it available that would be great
- Andrew: the only other thing I thought of, is in some way though the assumption is that the mechanism to get public key for the issuer would still use the JW KS configuration
- David: I believe so. I like that idea.
- Andrew: agree, it is quite tied to JWT and JWT mechanisms. It's just the format of it is not quite a JWT.
- David: They diverge as a design philosophy.
- Andrew: okay so just have a bit of explanation in spec. take away double base extra 64 encoding and additionals, get all bonus of encoded actual string where you save bytes as well.
- Craig: suggest we do have a github repository so Jeremy Adams could accept community donations and manage when adding additional languages to have a test bed, so could there be some support from Susan's group.
- Susan: in long run most of standards would work towards Starter kit, minimal basic implementations and they won't be maintained as things ready to deploy, expect people to have to customise their own implementations and also test beds exactly when and where we can fit in the list of things to be done. Passports would likely be near to the front of queue where there is interest in moving forward. Also looking to produce documentation and feedback from people of their experiences to get these standards deployed.
- Reduce the level of pain for those trying to pick up subsequently even if painful for initial adopters.
-
- Craig: We could add a few lines of code of the library where we fetch the keys and even have them expire, common in security realms to have keys and refetch to see if they have expired. Have this in the documentation, NIH might choose not to do that, may not want to refetch the keys.
- Susan: could we look at a implementation guide, this is core stuff to meet the standard.

Moving onto Mikael's questions

- Is visas had come from the same issuer but were visas about different things, broker could not merge them into a single visa entry.
- A weakness of this condensed method, that if the client wants to downscope the passport and drop some of the visas from the condensed passport then it needs to consult the visa issuer as it doesn't have the resigning.
-

- Craig: it's something we could look at. The visa issuer we would expose already a way to fetch the keys, as part of what we discussed. Maybe there is another end point that visa issuers if using this mechanism they offer an end point much like a token token point if take in current visa with the signature, bit like a basic bearer token off. Pass it in and request a subset yourself without even the broker involved but then need a broker to wrap it. Could be the broker that is doing this, broker collects various visas and the client only knows about it's own broker. If end point exposed the broker could downscope and the broker doesn't need a direct relationship. We could ask visa issuer to expose an end point if we want to solve it generically. It is an area that we can explore. If we do it immediately or not is the big question
- Andrew: certain type of broker getting a nightly dump from all around Australia, it may be something that gets dumped into the broker's system so the mechanism gets broken, just a floating idea.
- We haven't defined how brokers would talk to visa issuers in particular, that might be a mechanism by which it works
- Craig: that is the nice way of not specifying the relationship between broker and visa issuer that it lets it work with any architecture.

- **Craig: shall we leave this aspect for now and say this is a weakness that we should address at some point but not a blocking issue, can we move the entire proposal forward? Or is this a blocking issue?**

- Andrew: worst case, the issuer you could have an array of those and allow someone to put the issuer in twice. Same issuer to appear multiple times, devolve back into a multi visas from same issuer pattern at expanse of multiple.
- Mikael: introduce standard way for clients to request downscope passports directly from the broker. Condensed format can be one component or where it come together for the solution for direct request or authorisation request so the master passport is never exposed to the client. =
- Craig: yes it is the intention that this would be used with downscoping so can limit number of entries are in this condensed passport. If a researcher has access to 100 things you may only be using 3 at a time so get a small passport here. These would be complimentary. Maybe be required to not exceed 4k that we need to downscope. This is a downscoped passport.
- Kurt: yes agree
- Susan: Mikael, does that go some way to address your question?
- Mikael: Yes kind of does, but other way to request downscoped passports from the broker directly.. Trying to understand to which direction this is going and what can we expect, maybe this condensed passport to come together and request a particular resource from the broker right in the beginning.
- Craig: are you talking about the authorised end point?
- Mikael: yep from the authorised end point. My concern is that we need to trust the client if the client receives the full passport or master passport and client goes and

downscopes it by requesting a new passport from the broker or directly uses the self-contained passport and drops some visas that aren't necessary, then we need to trust the client and it's an assumption. This may have its own problems

- Craig: client not issuer but client trusted to hold that route passport token and to contact the broker to allow that downscope and if that leaks and other problems happens we need to trust the perimeter around that client.
- Kurt: it is a goal of some systems to know every participant and to have some trust relationship, this is where the chain of trust is most applicable here, so one authority doesn't need to know everybody but some authority that does. However in this case, almost the definition of what it means to federate is to allow unknown participants in the flow of processing data at some point, don't know it will be possible to know everybody in all cases. In some cases we need to lock that down, we are looking at this in NIH too. For general case don't necessarily to know every entity that is going to be touching data or involved in the flow. Don't know it before you start the processing. One of the results of the flow we are looking at.
- Craig: too complicated to know everyone in the network. I think it is also compatible for the client to issue downscoped passport right off the bat if known use case, can authorise and get something downscoped for its use case and go off and use it and never touch a route passport that has all of its visas/keys. Those aren't incompatible but might be a choice on clients behalf. Have a mechanism to get full passport and add end point/way to get 4k passport back immediately as a subset, maybe different clients and brokers have different requirements.
- David: that is a key point, that there has to be way more than one way to do certain operations. Know we want one spec to say one big bold statement but ultimately different clients will operate in different ways downstream.
- Craig: Try to keep it from diverge too strongly
- Craig: having a couple of end points that use passports seem ok
- Susan: any other comments here?
- Mikael: looks ok for me
- Susan: it seems like we are in agreement on looking at this 4k passports idea. Concerns on original authority and concerns that our system would be interoperable with OAuth and OAuth2-like systems. Feels like we are comfortable that this model can satisfy NIH interest in retaining original authority. What about the other side?

How will this system work with existing infrastructure?

- Craig: we need to keep working on the details as we move towards a PR let's set that aside and assume that we have such a thing
- How does this fit in with the previous conversations around an OAuth2 model or not
- How does authorisation work with internet services or web services etc? This smaller passport would fit with most APIs and security systems and infrastructure looking for a bearer token, can be attached as a header, specify word bearer and fit a token like this. Processed by systems people use here. One of the big advantages of this 4k passport.
- What end points are you hitting to generate these passports?

- For now the end points that are working on allocating passports not the usage. Should they be OAuth2 type allocations of passports and managing of these tokens or should we have more custom end points? More aligned with the first major issue on the list.
- Susan: my interest is that I had heard previously that we want the end product to address and satisfy. Is this capable of addressing the advantages we identified as using existing infrastructure?
- Craig: high level of main key issues as before and we can have a discussion on how well it may or may not fit.
- Kurt: I'm fine with what's being said (wrt original authority and 4k passports specifically)
- Craig: Main thing before issue 1, "mint new tokens whilst retaining Original authority to Clearinghouse." 4k passports allows us to keep original authority whether using a custom or OAuth end point, would be able to issue a token that would be compatible with what OAuth 2 would expect the token to come out as. So I think the answer here is yes.
- Kurt: not sure where it came from that there are a number of things that can't be considered in isolation, need to be all solvable. OA is one thing but ability to do this and the refresh mechanism is all bundled up together. As soon as saying things are in an OAuth manner then we get into what that means (i.e. non OA).
- Craig: yes you are right, if one of these things doesn't hold may or may not make sense to use OAuth2. If this is the format that gets passed back from various end points, there is a way to keep that authority, it's not as though you throw it away when generating a token.
- Kurt: embedded token is an object that requires a signature to be verified, it;'s not required to be a JWT. the format we are exploring says there is embedded data that is a unit that can be signed and preserved. That's the requirement behind it.
- Craig: the outer token of 4k token is a JWT but the content within is happens to be signed to keep that OA, but the outer token of it is a JWT. 4k passport is more like an access token, we could use this like an access token but whoever implements it is not looking for scope (most popular way of using OAuth2), but not only way to do authorisations. So need to add extra code that looks for visas, and accepts authorisation here. Could be used as a bearer token to get access to various different services.
- Andrew: the original passport access token that comes down in passport 1 explicitly doesn't have GA4GH claims in it because potentially they are too large (David: yes correct), now suggested a smaller one theoretically they could come down because downscoping mechanism. Original premises that access token of first interaction you have normally didn't have GA4GH stuff and that's why we need to go to user info and other end points to get the passport. In a world where everything is smaller could give master passport.
- Kurt: but you are not achieving anything other than conflating things that should not be conflated. The OIDC access token gives you access to user info which gives you access to PPI which cannot be part of the passport why it can't also act as the

passport. You don't want the token that fives you access to user's email address that also is the token that gives you access to the data.

- Andrew: I find that a reasonably weak argument, it's got someone's email in it.
- Kurt: email and their profile, any of the things can be turned from user info according to OIDC spec, nothing off limits, name, age, address, sex, photo. This is not the appropriate scope, the mechanism that can be originally designed you go through OIDC to get a token that gives you access to user info and one thing pulling from user info was a passport. Bit of confusion that resulted from that that we are trying to address by saying that's dedicate an end point to it but once you get a passport it has a different use and it should not be able to be used to gain access to personal information about the authorised user.
- David: is it just a statement about this be over used for personal info. Think we agree that we need a separate end point
- Kurt: the point is that the OIDC access token does not have visas, it operates on basis of scope and this is more access than what is included in passport.
- Passport does not act on basis of scope but operates on basis of embedded visas. why are we rushing to put them together?
- David: originally kept them separate as wanted a pure OIDC mechanism that would be compatible with the general world in doing JWT access tokens in headers to access user info, known and understand and acceptable flow. We all have stuff written for it, then when get the big claims you need to move away and diverge. Originally had them go into user end point as that is what existed. But separate end points are acceptable too.
- Craig: scanning Kurt while you were talking for the issue of not wanting to include certain types of information. Maybe not well captured in this table so will stop here at offline tokens (number 11 in the table). A lot of the user info claims like email and profile, released only if you have appropriate scopes on the access token and so if this is a concern we could state if the community agrees then don't release these, don't return these claims by user end point and scopes not included in part of this token. Or say depends on your use case, if google were to accept these tokens in healthcare we probably want to have third party tokens that are fairly open. We wouldn't necessarily want to limit what scopes could be in there, but maybe NIH would. If you call this a passport end point then stripe off the email, profile etc.
- Then simply not allow you, you will get nothing or only detailed visas whatever is reasonable.
- Some combination of specs is entirely possible and not entirely incompatible.
- Susan: this is useful for us to have this conversation to make sure we have this shared understanding, know there are people who have been in this conversation for different lengths of time.
- Andrew: put this in spec as otherwise someone from OAuth world wouldn't know why you wouldn't put in an ID token, this needs to be explained in the spec as not obvious.
- Craig: we could explain and allow people to limit what gets returned.
- Max: OpenID is the only mandatory scope

- Kurt: the way people get their passport initially is human interaction, will access your personal information. You as a user give your consent for that to happen. It's the downstream operation that doesn't have consent to have access to the user, which is why you can't make it available through the token.
- Craig: maybe we need a mechanism, it is limiting to say you can't have the visas there and have user info at the same time as there might be use cases for that. If a certain scope is present and use this as an offline token or some other scope that you need to use, you must stripe off all those other scopes, and you no longer have access to all the user PPI information.
- Andrew: do we intend that most visas involve downscoping, could this be where it is stripped?
- Craig: but might be use cases that you don't want to touch your route passport, i don't think we should necessarily disallow that. Very least, we need a mechanism that once put in the workflows, maybe we are able to say in the spec that there will be a special way to signal that and must not return that extra information. At least allow NIH to do that for its use case.
- Mikael: also possible that a service would expose a web UI for the user has request to broker, one with profile and email scopes which is consumed by that service itself and requests another access token that only has GA4GH scope, that's the one that's exposed to downstream APIs.
- Craig: yes we could manage with scopes or other mechanisms
- Kurt: separate the idea of size and downscoping while maybe an effect of downscoping you may find a smaller token that isn't the purpose of downscoping, the purpose is to reduce the amount of authorisations that you have in it. Do not conflate size and downscoping.
- Susan: By the time we reach a downscope passport that it will be so downscoped that have we already reached a minimally downscoped passport by that stage? As we talked about a upper limit of visas in this?
- Kurt: upper limit of permissions in a passport is a technical detail, not a detail on amount of authority granted to an individual. Could break them into multiple tokens, a purely engineering issue.
- Craig: think there would be a desire to further downscope, purposes where you would continue to downscope depending on the services you interact with. E.g. handle 20 datasets but next operation only need to use one dataset so get another passport with one dataset on it. Contact service with just one dataset, but never get more access. It only goes one direction. It's not just security it's also privacy. Want passports to contain only what is necessarily for the flow (GDPR etc)

- Max: **I have a point that we need to discuss compatibility with conventional systems with 4k passports.** If this will have an audience in it's JWT and the role of this would be? Downscoping is good and will probably remove a lot of opportunities to send a token sent to one server to be replaced by one by another server where this requires totally different visas. Think there will be cases in real world where some resource servers as part of loosely affiliated servers that use similar visas. One

visa for many different places, without a token having some kind of **audience** it would be replayable between different resource servers.

- David: real world thing- for that exact term, e.g. between broad and UoC, enforcing mutual TLS as mechanism for holder of passport is authorised and can be more trusted from a security aspect. Replaces the idea of audience.

- Max: but here there will never truly be third party systems that researchers can use to access data

- Kurt: no the model supports the case of data sensitive constraints to who can access. If you are serving sensitive data in that case, access to data may require knowledge of connecting systems so you understand the distribution of data. May require closed networks for data management.

- Mac: if the system has an audience then may be usable for less sensitive systems. But without audience you need some other mechanism like mutual TLS.

- Kurt: what audience really does, does nothing to a rogue end point that receives a token as it gets ignored. Voluntarily observed

- Max: not about the token being received by a r server it's about other servers not accepting it.

- Kurt: we can put that in and dozens of others and it doesn't do anything other than add bloat to the token. Where audience useful and insisted upon is that the OAuth2 system is used to grant the authority to masquerade as an identity.

- Max: but audience usually referring to a resource server not a user

- Kurt: OAuth 2 model token says here is the subject which is something that can be changed along the way and here is the audience, at resource server means this was intended for me issued by so and so and make sure of the count for the subject

- Max: I would disagree, don't need a subject in a token. You can orthogonally say passports don't need subject and audience still a valuable security mechanism

See FASP document link for discussion on mechanism of audience: Re 'aud' (audience), see also: http://go/ga4gh-fasp-security

- Craig: we do need to understand how do we avoid having this proliferation of data that you can leak almost entire network of GA4GH with a rogue actor. Audience is once way that. If we don't need audience we need another similar mechanism that prevents this type of attack. We need another mechanism.

- The outer broker also has a mechanism for this, so need a little bit of trust in the broker so makes sure it doesn't capture and rebundle visas. You can't trust any broker that comes along, so they don't take replayed visas.

Summary of our discussion

Useful to capture the follow things in the design doc:

e) Discussion of use of user info
f) Maintaining the whole passport
g) Downscoping
h) Audience mechanisms

2021-08-12: Proposal of a smaller 4K passport

Chair: Susan Fairley
Attendees - Name (Affiliation): Kurt Rodarmer (NIH), Max Barkley (DNAstack), Susan Fairley (GA4GH), Alice Mann (GA4GH), Tom Conner (Broad), Andrew Patterson, David Bernick (Broad), Craig Voisin (Google)
Design doc: https://docs.google.com/document/d/1lSRIJRFSIB8EMww_yOY6hWkT6O7jDABdmxegsmFBD24/edit

Summary of our discussion

- We reached agreement on exploring the 4K passport proposal from Craig and Kurt

Minutes:

- Last week we were working our way through the design document, discussed the first point and summarised the position we reached but we hadn't reached consensus on whether one or the other two solutions were superior to that point.
- One option to move forward, resume the discussion from last week, a little hesitate to recommend this as it feels there has probably already been significant discussion on that and may not be helping us to resolve things.
- Interesting to look more closely at where we do have alignment and agreement so we can get to something a little more concrete to the things we have in common, and take this conversation out of this group and back the wider world.
- Something coming through is that we seem to have different contexts and scenarios in mind and different levels of trust

- Some scenarios that some people are happy and relaxed with token exchange option and some where people are less comfortable with that
- To what extent do you trust somebody? Probably needs more of the people from a policy mindset. Concerned we are having this discussion in relative isolation from them.
- What levels are necessary in trust? We also need to take this back out to the implementers.
- Produce a document that summarise a document to say what our options are and is there a way to make these two things together.
- Happy to hear people's responses.

- Max: sounds like a reasonable approach to me. How can we advance that in particulars? Would use cases be a good place to start? Where we are coming from with the usages we imagine.

- Susan: let's get down on paper, we need a standard does passports. Assuming that passports already has use cases. Get a couple of different implementers or approaches. Do we have more options that just these two? Is it really a binary choice, what are the overlap between these two choices. For me, security, adoptability and interoperability a concern. I worry that we are potentially in danger of getting caught on two things when there may be other choices and secondly there are components of this conversation that need input with other people. Ultimately this decision will rest with people at RI and policy point of view, what's necessary for somebody to become a trusted participant in this scenario.

- We need to find a way to bridge that conversation with other people, lose JWT etc other acronyms as meaningless to other people. Possibly need to connect this to the wider GA4GH world. If this would go in as a new standard, would need approval at SC so they will have a say there.
- Conversations here so far are incredibly valuable and appreciated but not going to achieve anything with current phraseology in making ourselves understood.
- Craig: agree, hard for anyone of us on the call no matter how technical we are. Haven't previously though about it in terms of multiple audiences so maybe we need multiple documents. One that removes a lot of technical jargon and from a policy point of view, this is the consequences. It may also be useful to pull together a slightly more technical version of the doc, but have a nice tight description probably based on requirements. These are various requirements and get a feel from technical people what the requirements are.
- Susan: issue of different levels of trust, Kurt sharing info about NIH's view on these things. NIH major stakeholder but need something that will work for everyone. Maybe we need to acknowledge this isn't a one size fits all.
- It feels to me from what we are hearing is we have this fundamental difference in needs from different stakeholders. Useful to distil this down and then can be shared with likes of EGA and GEM Japan. How do they feel about this? Stakeholders who

are using passports. Practically impossible for all to be part of this conversation at the moment.

- Tom: less is more and having one document is good. It's overwhelming to have so many different documents. Thought that understand that we want more policy folks to understand what's in the spec. what if one document with an introduction that states the policy intent and then technical details in the same document. Folks are free to delve into it if they want.

- Andrew: sometimes in standards doc they feel they just have the standards detail, but I feel the standard is lessened because it doesn't have the explanation or context at the start. Someone writes a paragraph at the top to explain the rationale, even if that is non-normative. I find this much better. Need information about how they got to some of those standards.

- David B: historically speaking this was really simple! But it has got bad since.
- Susan: agreement on going ahead with one document.
- Mikael: would it be useful to have a separate use case document? Introduces the use cases and states the problems that need to be solved by passports. Use cases and then reflect back to that document, when we propose those solutions. If we don't have use cases then it becomes difficult if we know we have solved the real life issues.
- Susan: what do we already have as use cases?
- Craig: I don't think we have a succinct version as a consumable? Of here are the main use cases.
- Susan: another point made to me was having a good analogy. One was suggested to me of the concept of covid green passes? Do you trust the nation state and their vaccination programme? Is there a potential to find a useful working analogy? Use cases should be the driving thing of what we are doing. But may lose people if use cases get very detail.
- David B: this was something we thought about, the spec would be bundled up with supporting documents about how to use it, but we discarded this if spec not clear enough and the use case document diverges from the spec. but I think having a bunch of use case documents are just as good as the spec, but we should make sure the use cases and spec don't diverge which was the issue before.
- Craig: even policy people who may not be well represented on this call, these use cases should be readable by a policy person.

Only official higher-level passport context/use cases is here:
https://github.com/ga4gh-duri/ga4gh-duri.github.io/tree/master/researcher_ids

Moving discussion on

- Should we talk where there is scope for alignment? Or where we are aligned?
- Is there any issue for anyone that we should discuss now?

- Craig: yes on one such issue. Kurt and I were working on on Tuesday about that could get feedback on. It might be helpful to get some feedback. Kurt and I were editing this.
- It's a way to discuss Fat passports. It is an underlying premise driving some of this work, is that we need to have these self contained passports that can be quite large.
- Kurt and I started documenting why NIH needs such a thing and other members of the community.
- We had two bullets and added a third to the docs.
- more importantly background, there was an alternative working on how we can get rid of having large tokens and yet still meet those requirements.
- It may get us more aligned than we are today. Do we want to keep going in this direction? Or should we back off from this proposal, we would love to get to this point today.
- Susan: this sounds interesting so let's take 10-15 minutes to do this.

David B

- Historical note: the fat passports emerged from the realization that passing around the oauth access token jwt wasn't advisable and passing around the derived claims was reasonable. The biggest problem with the result was size. It sounds like craig has a potential solution there.

Craig and Kurt's possible solution:

- Below the current table in the existing design doc: why a large self-contained passport is needed?Background is in the document
- What's the alternative to large tokens: compromise required of course
- What if we had a new visa format? Visas tend to be large, as you start getting multiple visas it eats up multiple K, tends to break most authorisation in terms of the way the internet works, to be compatible with a lot of other standards and you end up with large tokens that don't fit in headers
- What if we try to change the standard so we try to have a more compact representation of visas and then also accept some kind of practical limit to what you can fit in 4K object. This is generally what fits in most things on internet and technologies and components - still be a little bit more aligned with other standards.
- 4K modern practical limit and back off JWS string, encoded string that actually uses a whole bunch of extra bits so don't have 4K have roughly 3K for content. We could still offer this call back option to fetch more info or visas, maybe this isn't required. NH could say we don't want to do that.
- You could specify a bunch of things but once hit the cap, that's the most you can do. Limitation of what this would support.

- Susan: recap, this is a way of making it smaller. Size of overall passport would be reduced. On top of that the element in the passport that would hold the visas, we are replacing a cap on how many of things you could have in a given time.
- David: Size has always been the bug bear. That has forced into one direction or another
- Mikael: is the idea that the condensed visa then added to the access token.
- Craig: it's right inside self-contained passport, have the visas in there and still keep core things that our specs do well, retain the original authority. Still signed, have enough info for a clearing house to make a decision if this person has authority. Try to do this in a compact way.
- Would be a practical limit on how many things you could do. If willing to accept these limitations.
- Well aligning more with the way other systems do things, which is not make it this giant token.
- Passport still is a JWT
- Unencoded piece you have roughly 3K. use some of the 3K on JWT stuff
- Header has everything the broker needs, algorithm and key etc
- Then you have the Payload
- Then signature of JWT at the end
- Outer overhead, around 500 bytes if include standard claims like "iss", "jti", "sub"
- GA4GH claim, similar to a passport claim today "ga4gh"
- Structure after this, version number "vn"
- "iss" issuers, "<visa-iss>"
- Details later, get a feel for if this is a good direction
- "v": what your visas actually look like. EGA visa may look like c:01234 a controlled access visa.
- C: indicates controlled access visa
- Up to visa issuer pick the smallest way you can encode your dataset. Phs number is normally an NIH number. NIH could decide to drop phs?

Kurt:

- The ONLY valid "issuer" claim is indicated by the signature. If you don't recognize the signature, the rest is useless. And if you do recognize (and honour) the signature, further 'iss' declarations are superfluous or incorrect.

- This might represent a whole dataset and subsets, NIH could come up with a system of compressed dataset to make as short as possible. These strings typically 10 or 20 bytes in size to add a visa in.

- Susan: subsets of things, I would have imagined many groups who have responsibility of large datasets would be interested. Has this been part of this discussion before? Do we know if they can classify things like this? Have we consulted already?

- Craig: don't know how much consulting we have but the current specs do allow any issuer to figure out how to encode subsets, so in the current specs you are able to do this but it isn't well written down. More we can do in this regard.

- David: In practice don't people like using iss to check the well-known site for some (perhaps imperfect) verification?

- Craig: main idea that you have a very compact form. These things can be repackaged through brokers etc if we made this a JSON object might be more bytes than here. As soon as wrapped in a string so if somebody reformats and adds extra space of line feeds as opposed to most compact way to encode you don't want to invalidate the signature below, most code and most things that manipulate these things wouldn't accidentally break the signature. Efficient and helps with signature retention as re-wrapped across different brokers. You can still have expiry from visa issuer point of view not broker point of view. We will keep a key that the visa issuer is using the sign. Then signature block so we will try to keep the small ones, elliptical curve. Keep this as small as possible whilst offering really good security. You could have multiple visa issuer and still put them on here. If this was a 1k string, average size was 10 bytes, you could have 100 entries in this string, if 20 bytes in length then a little more detail here maybe you are limited to 50 or 40 entries in a 1k allocation. Any one issuer can allocate quite a few things, if multiple issuers, overhead here to allocated a new block so are using up more bytes. But if only had one or two, subset only what you need for execution for compute job you could still have 30 datasets across NIH and still have enough room to have visa issuer from other locations and still keep overall size to a couple of K in side this block. That's the basic premise behind this. Place a limitation on this and looks like a regular bearer authorisation token that most of the internet uses and this may bridge the gap with other tools. I know there are compromises but is this an acceptable compromise?

- David: this is a derived token, oauth access token, that goes to user info but do you go to a new end point? Where does this come from?

- Craig: this token does come from some end point on the broker. Discussion, what is that end point? This is not when you first log in, what you get. Route passport has all of my power, hit an end point on the broker to give you one of these things that you use for execution. Issue access right inside the token

- David: we need another end point on the broker then? is this token the oauth2 access token or a "derived" token gotten from an endpoint on the broker? (Similar to userinfo or IN the userinfo itself)

- Craig: we can still debate do we fit inside the OAuth 2 totally etc. At least from a client point of view we look more like a regular token.

- Max: using C to encode a controlled access visa type but what are your thoughts about other visa types? For ASI will care a lot about affiliated institute and role for example. In current spec nothing to stop you making up your own visa types. Do we need to standardised short form encodings for all visa types or need a mechanism to say what the compression is? Would we need to standardised short code for all visa types.

- Craig: we would say what all one letter code is as standard setters. See a = affiliation and r = role. We would define these things. If want to use your own custom visa type, if the descriptions don't fit what you want, still make it a compressed format. The underscore character is the extension, for that issuer, you don't need to namescope this for a url. For that issuer you can define your own characters that have your own meaning and use cases. Any characters not starting with underscore are reserved for our specs

- Kurt: the only trustworthy piece of info in terms of the issuer is the signature, once you've accepted that, you know the issuer, if need to use other claims inside to identify an issuer then that's a problem. Craig has here a means of grouping visa issuers separat from passport issuers.

- Andrew: to what extent does the beauty of JWT is there's a 1000 libraries that will mint and decode and verify, to what extent would we be able to leverage those kinds of libraries to do these signatory/verify signature stuff?

- Kurt: there would be no off the shelf JWT libraries that would provide that signature verification

- Craig: that is a limitation of this approach, the way it is wrapped it would not work with most libraries, need to write extra code to do it. A lot of libraries do let you check signatures just not the whole JWT. Answer is somewhere in between. Probably are libraries that let you check a key signature. How you get the public key, need to do that yourself

- David: doesn't require a huge amount of work, not too much of a departure of what passports does already

- Susan: we've spoken about it being advantageous to use existing infrastructure. How different is this from the current implementations of passports and is it a big ask to move from current implementations?

- David: if the current way of using passports then this isn't a departure. If using off the shelf resource server, this is already different.

- Craig: custom work is required, even with the current spec. other than the point that Andrew brought up as to how you call the library and checking the signature it's not a huge departure from version 1.0

- Kurt: for passport token still standard JWT, could use standard off the shelf to process it. Once you get to the claims which are the visas then you everything is new, no longer have an off the shelf means of validating the signature on the visas, you have to code that yourself, using off the shelf libraries though. It would change all of the parsas for anyone who consumes visas today who have based their parsas on embedded JWTs and JSON model. But this type of token gives you much more compatibility than the fat token. For this can have backward compatibility for sending opaque token through the http header, with the other one you get more off the shelf reuse for processing the token but you can't send it in header.

- Mikael: observation in current approach each visa is signed individually by the visa issuer but here several visas are bundled and signed together. Current approach visa downscoping can be done by the broker and the broker picks those visas to downstream but here because condensed visa signed by issuer, it's the issuer need to sign if its downscoped.

- Kurt: in our case it's always like that, only NIH can downscope the visa. We don't have permission in a separate visa/

- Craig: separate signed visas it is possible to for broker to do it itself. Mikael's point is true.

- David: Side note: my personal opinion is that we shouldn't be scared of making reference software anyhow.

- Tom: I was just thinking that we should publish an open-source parser implementation in at least one if not two languages. The code would be small and it would be open and easy to read all the code

- Kurt: NIH is publishing its software for this in usable form.

Is everyone agreeable with exploring this new direction?

- Kurt: I support this direction, we've had this discussion for years, opposition from people who want to use full syntax to represent any information, its extremely wasteful. We have the experience of creating large objects in the past. Economy of representation is important. Fully supportive of going to a compact representation. Benefits, not using base 364

- David: it's a good direction. The size thing is what pushed us away from this in the first place. It's good to come back to it

- Andrew: base 64 dots inside dots felt strange. Initial worry is accessibility of cryptography libraries to do work

- Tom: love this direction

- Max: I want to see what it looks like with other claims, promising direction

- Mikael: potentially yes but depends how it fits in the wider context. part of access token or part of self-contained passport? These details are missing.

2021-08-03 : Pros and Cons for each of the two alternative options

Chair: Susan Fairley
Attendees - Name (Affiliation): Kurt Rodarmer (NIH), Max Barkley (DNAstack), Susan Fairley (GA4GH), Alice Mann (GA4GH), Tom Conner (Broad), Andrew Patterson, David Bernick (Broad)

	Actions Arising	Assigned To	Deadline
1	Continue the discussion for the next meeting	All	
2	Gather the requirements we have discussed so far	Susan and Alice	
3			
4			

Minutes

Minutes Summary

- We continued discussing at length Problem 1 and the two alternatives for getting a passport token.
- We added pros and cons for each alternative into the design document.
- Clearly there is a split between those supporting using OAuth at the start with a later diversion and those who want custom from the start. We need to understand wider requirements to be able to choose the appropriate option

Full Minutes

- **Susan intro and summary: Work towards a document that we can circulate widely, something that is in plain English, outlines the requirements, what the options and trade-offs are and ultimately makes a recommendation.**
- Discussing item 1 in the table last week, got to the stage where we were pursuing a discussion about what the requirements are how can OAuth2 could do this, and what would that look like?
- **We finished last time with discussing item 1: minting new tokens whilst maintaining original authority at the clearing**

- Max: I will summarise my understanding of where we reached. We have priorities that Craig curated with priority 1 (highest), P2, P3 etc.
- P1 is maintaining original authority to the clearinghouse. That is the first point, how do we maintain this. We had scoped down the discussion as there are a lot of different parts. E.g. where do you get the JWT that has these visas in that maintains original authority. But also how do you use that downstream to actually get resources? We had explicitly downscoped the convo to talking explicitly about how do we get this JWT that contains visas? And what do you do with it afterwards? Previous comment: different options for when to switch from passports to downstream mechanisms i.e. where to place the clearinghouse and what comes after.
- Kurt: at least in talking with Craig afterward, we decided that the placement of the Clearinghouse in the flow does end up being critical. So I don't think it can be eliminated from the discussion.
- Susan: yes we later commented on the document that this should be in scope.
- Max: we had focused on the preceding part without talking too much out of this. We can step forward in a couple of ways. Let's make sure we are aligned in the way we decide to get the token that contains the embedded visas. Or is there a particular blocker to coming to a concrete decision on this that is in the previously out of scope area that needs to be discussed.
- Max: put forward to the group, reaffirm where we are? Can we make a decision or do we need to go deeper on how do you get a JWT that contains embedded visas.

- David B: both my original vision and my 1:2ish vision, normal OIDC/OAuth2 workflow happens and an access token is gotten and used on a user info end point or some end point on the broker to get a JSON blob that can also be a JWT. That created some confusion. There needs to be a way to get just the JWT of that user info blob which contains the visas. I thought this was uncontroversial, access token go to an end point and get a representation of the visas. The big question is then what do you do with that stuff?
- Max: my understanding, we all agree more or less on those steps and the only nagging details, is that a new end point or is there an existing OAuth end point that

can do this that can exchange the access token and getting the JWT with the embedded visas. Readily concede that this is probably one of the smaller decisions on passports that need to be made.

- Kurt: End points are cheap, just names in the server, same server and the same service that is handling them for the most point. They are entry points, purpose of them is to distinguish their functionality one from the other, not to provide one end point that is parameterised to provide 100 different functions. You have then 100 different end points for that many functions. Overloading user info end point only made sense if getting user info. Overloading the token end point doesn't make sense in the same reason that there are semantic differences. We are talking about taking an OAuth token and asking for a JWT, which is a passport token and there are other things that we envision to operate on the passport. It doesn't make sense to overload existing end points for what purpose? I don't know what the purpose would be, no existing software could utilise end points in that way. So you haven't done anything other than introduce confusion in the spec. It's very clear that If introduce a new end point, this is the end point that uses passports. Something that is not documented elsewhere, documented in GA4GH, we don't need to try to say we haven't stepped outside OAuth spec but introduced functionality that no one can use as still need custom software to use it.
- David d: Do we take on the burden of producing something new? Including dealing with organisations saying they need to deal with something new.
- Kurt: new no matter what so if it's new and confusable and new and clear, those are the only choices.
- Max**: just to be clear of the two columns, the alternatives. Before and now discussing completely new end point that you send a request with an access token and get back these fat JWT with embedded visas or some kind of OAuth extension** most likely a token exchange but some other grant type extension.

- Kurt: token exchange runs into difficulty with preservation of authority and that's something that we can't keep mixing the two, concentrate on one variable at a time, otherwise we'll be misled.
- Susan: Max put forward the idea talking about point 1 or if we want to consider whether or not the territory beyond the CH is considered as part of this discussion or we focus primarily on point 1.
- Kurt: for point 1: clearing house not yet in the discussion. CH does not yet have anything to operate on, how you even get to the token that the CH works on.

- Max: **love to get a shared understanding of what exactly the two alternatives are**. Either way you go it's one http request that you send with one token that gives you back a JWT with embedded visas.

- In either of these approaches you repurpose an Oauth end point or you make a new end point. You are discussing two alternatives where in either case it is an API for

sending a single http request with an access token and you get back a JWT with embedded visas.

- Max: only point for why we should consider an OAuth token exchange or extended grant type is what happens downstream? What is the thing that ties these APIs together? What are the conventions at GA4GH are we advocating for as a whole? Maybe think about Token exchange as it is an OAuth extension, so if further downstream we use more Oauth, it is consistent with that and things more upstream from these things,
- David B: from a marketing perspective, how do we demonstrate that adoptability is not a huge burden.
- Max: I'm in camp OAuth, if we could hypothetically get a survey of responses from developers who would use this standard and show them two drafts of the API and they say I find it better if a new passport end point then in the greater scheme of things then that's what we want to do and probably doesn't have a huge impact either way. But what are people going to adopt more easily?
- Kurt: problem is the security. Security is not a customer driven design process. First order of business is making is secure and second order of business is making it usable to people. People always think it is customer first.
- David B: but it's all important. It's important to have things useable and transparent or will be worked around and not used as a security tool.
- Kurt: marketing a different thing.
- Tom: I think we all agree that about security. Preserving original authority is absolute paramount here. Keeping authority is critical.

- Kurt: Right so to David's point. OAuth is 3 things, a convention and something like a protocol based on JWTs most of the time, it is a religion and a marketing slogan. It is not secure though. Designed to improve security in a certain context, when outside of comfort zone- the security it falls down. The hype around it is the opposite. In GA4GH before in this discussion, claims that if it's OAuth, confident we know it's secure. I can prove that beyond a shadow of a doubt that it is not. As an NIH representation those issues are a concern.

- Tom: Oauth does get used a shorthand for magic security dust. Within oauth many different flavours and grant types. You have to say more than just the one word oauth, big fan of following in the footsteps of others. Doing it in a well proven way is the best way to go for security sake.
- Kurt: design of an ecosecure operating system I worked on almost 20 years ago. Capability system, capabilities are essentially tokens. It becomes a way of life, yes following proven security models like a capability model is a good thing to do.
- David b: but adoptability, given resources at my disposal, don't mind having something divergent of the norm. how things get done that are future facing. Federated authorisation has been attempted by many people and has been unsuccessful. This is what we are trying to do. Do we need new paradigms here and

we have to accept that anyone implementing it has to have off the shelf methods and we have to accept that GA4GH is responsible for implementing new software? That's what I keep coming back to. I don't mind doing that as Broad has those resources but not everything does.

- Max: make up of how these things are considered changes when focused on a single request. Single request where exchange access token for JWT with visas. Both of end points are almost indistinguishable because they are single requests.
- Kurt: yes they are but except if later following capabilities model, we need the ability to downgrade the capabilities of the token, which isn't accommodated by trying to preserve OAuth capability, unless we go to token exchange model which is going to fall apart later in the flow.
- We are looking for an end point that manages passports and designed right from the start to swap back and forth between fat and skinny passports i.e. the downscoping of passports.
- Andrew: is your proposal then that there will be multiple new end points?
- Kurt: not necessarily. This is why I want to go early to having a passport end point. Dedicated to operations on passports.
- Andrew: previously you said we could mint as many end points as we want
- Kurt: that is true. We could go that route, don't see a problem with it.
- Andrew: either new passport end point or token exchange, it's one http request and we vaguely get to the same end point. Is this the juggling point we are currently at for this? As we go further through points.
- Kurt: seem to be making argument you could overload something that has an existing functionality and semantics and to change the semantics and provide additional functionality.
- There's a reason that these tokens don't participate in the oauth flow. Two main reasons are refresh model and the reissuing of tokens, resigning of tokens, essentially token exchanges as you go down authorisation certs. Those two things together make the oauth flow untenable for a GA4GH workflow.
- Max: I don't think anyone is proposing nesting token exchanges that resign. Would this token exchange for this one end point where exchange access for JWT containing visas, is that in terms of security, how is it substantially different from a new end point?
- **Kurt: if on the same server and signed by original authority, it is not different. That is a matter of naming. If concerned with names. Problem I have is that I look at it from other standpoint, there seems to be an effort to stay within oauth even though we think that after that we are going to exit oauth. Or is it a step along the argument path to stay in oauth the entire way, which I have been arguing for two years now doesn't work. Why? What benefit does it provide us to not use a dedicated end point for this thing that does not return an oauth compatible token.**

- Max: token exchange on this one access server, this is what we are talking about. Not more complicated exchange than that

- David B: we are talking about a departure at some point in this flow from OAuth. It's after a certain point, therefore moving away from it. That means that GA4GH will have to support software implementations. It's not just about spec at that point. It's about building software implementations.
- Kurt: NIH is also trying to publish reference implementations either used directly or adapted or point of reference.
- David: don't mind if specs are new. We need to make sure that this is something people can use without each person writing a brand new implementation each time. Need all the tooling that is normative in these environments.
- Max: agree, the existence of robust implementations is really important for adoption. To the point of this end point why I would advocate using oauth has some advantages, if we can say this ned point is this RFC with these parameters then we save a bunch of human effort in our standards creation, we can lean on the reputation of the people who have authored this EFC and use it to boost our own message.
- If we think all other things being equal, then we can use this to save time and signal boosting how this fits into a broader ecosystem
- Kurt: but the act of going from OIDC/OAuth to get your initial access token is entirely in the existing RFCs. The act of going from that access token to a passport token that is still an OAuth operation as operates on an OAuth access token. But what it returns is just a resource that has been accessed, it is no longer an OAuth token. Then the question becomes when we start to do further operations like the exchange between the interacting and self contained or downscoping, does that then go to the new end point? As no longer in oauth world.
- Max: the thing about this token exchange is that you tell it the type of token you have and the type of token you want. So my thought is that we could say use this RFC but the token type of GA4GH passport is some URN we make, that's the canonical token type. Then you can easily say describe a flow where you want to exchange an oauth access token for a passport token or want to exchange a passport token for another passport token that is downscoped. Think it can harmoniously fit into that paradigm. Channel the effort to defining whole end point to what is the semantics of this token type
- Andrew: no client out there off the shelf that does this flow. But fitting in with an RFC that exists takes out the extra bits of a spec that we have to write. This is the new grant type etc and doing as per RFC… defining headers, mechanics etc. if we want to do it yourself have to bring all of that into your spec. leverage all of that work that has already been done. That is an advantage. But get Kurt point, more of an advantage to declare that we are out of OAuth the whole way. Otherwise have to say we are branched out and doing the specs for all of these end points. I do go along max's vibe where we can leverage existing RFCs and formal documents.
- Susan: useful points made in last 20 minutes or so. One point that came through end points if on same server it's the same but perhaps not representative of what is happening downstream. Agreed security and adoptability is a concern. Gone on now discussing the pros and cons and how closely aligned to oauth workflows, none of

these strictly oauth. Tom's earlier points maybe using oauth a little meaningless as various flavours in there. Do we have any clarity on anything and can we capture anything in the design doc now?

- Andrew: also totally down with going custom end points as per Kurt. It's not the biggest decision if we want to move on then add some custom end points.
- David B: is there an appetite to make a separate RFC that is aligned with way OAuth RFCS are written and documented for our new thing
- Kurt: I think we should and update JWT RFCs to correct some of the errors.
- David: if we diverge then we need to claim that but everything diverges from OAuth at some point and becomes a sub category. All things are off shoots?
- Kurt: oauth isn't even as specific as saying its JWT, common its JWT actual specification is JWT for tokens and handful of API protocols. What we are talking about doing is almost indistinguishable from it, only question is what you name your end points and handful of other properties around the flow. Any of these things in isolation, oauth not so much of a bad deal. It doesn't hold up in the flow. We can look at token exchange but ends up needing to be a definition of what the behaviours and the brokers.
- Max: shall we jot down pros and cons (please see the design doc for these).
- Susan: both of options have some flavour of oauth.
- Max: we have added the assumption that security and practical considerations are largely similar between two approaches of how you exchange an access token for JWT with embedded visas.
- Kurt: pros for custom option- RAS project has lost many months over arguing over confused specifications. Clear specifications are the main product of GA4GH. Having some OAuth and then divergence will result in losing hundreds of thousands of dollars. We would be able to introduce a new definition of functionality that allows us to be as specific as we need. We don't need to describe how it differs. Leaves the existing end points pristine, as opposed to being altered to introduce a new behaviour.

(please see the design doc for the rest of the pros and cons discussion).

2021-07-29: Token exchange mechanisms

Chair: Susan Fairley
Attendees - Name (Affiliation): Craig Voisin (Google), Kurt Rodarmer (NIH), Max Barkley (DNAstack), Susan Fairley (GA4GH), Alice Mann (GA4GH), Jeremy Adams (GA4GH) Tom Conner (Broad), Andrew Patterson

	Actions Arising	Assigned To	Deadline
1	Alice to share notes	Alice	
2	All to read Appendix provided by Kurt on NIH view of flow https://docs.google.com/document/d/1lSRIJRFSIB8EMww_yOY6hWkT6O7jDABdmxegsmFBD24/edit#heading=h.bsln8po91wc	All	By Next call
3			
4			

Summary of main discussion points and where to pick up next week

- Token exchange mechanisms: can we stay true to OAuth but also meet the requirements of maintaining original authority?
- How much do we want interoperability in the middle nodes

To continue the discussion next week:
- please read the appendix as highlighted by Kurt on the NIH perspective on flow
- We will continue the exploration of OAuth2 options wrt our requirements

Full minutes:

https://docs.google.com/document/d/1lSRIJRFSIB8EMww_yOY6hWkT6O7jDABdmxegsmFBD24/edit#heading=h.bsln8po91wc

- Susan we need to summarise at the end in plain language, document that spells out what are the requirements, foundational workstreams, but what needs to be in place for this to be adopted and used.
- Outline the options and the tradeoffs with those options, also a recommendation
- Document for wide circulation
- And a consultation period on this.
- Craig: shareable doc really important, even the most technical of us can get confused
- Craig: driving through this table here, 3 main fundamental goals, and modified about what is out of scope, do we want to make it completely out of scope (different options for when to switch from passports to downstream mechanisms). Look at where the clearinghouse exists, where the clearinghouse lives might affect our design decisions. Value on this call, still focus on number 1 (minting new tokens whilst retaining original authority to clearinghouse).
- How can we build on existing standards or limitations if certain requirements for the first issue. This will help guide us to what the issues are or decisions.

- Interoperability between implementations, hopefully decisions on first two will help guide us.
- Issue 4 may be where the clearinghouse exists, so may become in scope in these discussions.

- Decision that the clearinghouse is in scope to this discussion. No objections on the call.

- Back to topic 1

- Strange to use user info rather than introspect. Still using the OAuth2 tokens in the current design. Major feature is not just adding simple strings as scopes or even simple claims that get added to a JWT token, we have these signed additional array of visas that retain the original authority. Keep that through to the clearinghouse.
- Is that OAuth2, ship has sailed, we are off that path.
- What is in the passport spec today as how you get the tokens is mostly aligned with OAuth, little idiosyncrasies.
- Max: Big choice if we deviate what you do with passport JWT once you have it. Biggest bifurcation with an OAuth paradigm or something different. Touches on where the clearinghouse is, who is responsible for accepting those tokens.
- Craig: still like to look at the path up to the clearinghouse. Are the tokens that the broker is going to issue, are they just a direct extension to oauth 2 or are they incompatible.
- Any security edits to understand oauth2s will need ot be redone.
- Once throw out passport token and you move to another token, we need to discuss what this means.
- Kurt: suggest once done that now outside of scope of GA4GH protocol.
- Craig: like we do with AAI, with our downstream tokens don't use them forever. Once we exchange we are outside of the GA4GH protocol.
- Tom: internal processing within the trust boundary will be outside the scope of whatever the specs say, once in a trusted execution context. need to make sure that security analysis done in the past can be applied to passports visas and any downstream tokens. To me, main thing is to make sure can do that security analysis and vouch for the security of the whole system more so than whether we classify as oauth officially or as classified as a JWT token. Specifics on security parameters, will we allow unauthorised things to be happen or not.
- Craig: requirement in the security parameters and what we need to do. Needs to be something that we can attach the process to, to create an audit, security audit how these things work and appropriate security perimeter around whatever we come up with. Dependent on if classify as an oauth2 or JWT token.
- Kurt": not the type of token but the process or flow, both processes talk about JWTs, it's the semantics around in an oauth2 process, two problems here the refresh token mechanism and the loss of original authority, those two things are what we are trying to address. Not separate issues.

- Susan: useful to have a working definition of what original authority means. If oauth2 not doing that, how does OAuth 2 doing instead.
- Kurt: idea of original authority, letter of king bears kings seal in wax that makes it to destination, that letter transcribed two or 3 times in between, dangerous for recipient to accept as unclear how original. Designed something that contains signature of authority all way to end. OAuth creates bucket brigade of letters, recipient has to trust that the transcribers in between didn't change anything and they are essentially accepting someone' word. Minting your own money. OAuth uses chain of trust. As long as the trust increases from one stage to next. Provided in identity providers in OIDC. It really doesn't matter if some entity says NIH says you have access to these data, you can believe me as I'm trustworthy. System of chain of trust designed so that if any component fails either on purpose or due to human error, the entire system breaks down. That's not the way you design security systems.
- What are the alternative options here? Tokens to clearinghouse?
- Max: how is it that I get a JWT that contains embedded signed visas. More questions about once you have it who should accept it do you need to exchange it for other things. First question prior to those and probably more tractable. Oauth approach vs passport approach a lot of complexities
- Help if we clarify the specifics. How do we get a passport JWT?
- Summary where we left off: alternatives we could say this is something, we want this token to be semantically different than oauth, pass in the access token from passport flow *doesn't contain embedded visas and you get back something containing embedded visas and we think this should be a separate end point.
- *What is an end point?*
- Ouath not bothered about what tokens are, extended over years to handle other tokens and so why not extend end points so it can handle this.
- Craig: if oauth2 if we can stay on spec, we can follow what is written in base oauth2 spec and follow the extensions as how to extend oauth2, then if it makes sense and it can handle what we want to do, why wouldn't we want to go with oauth2 as so well understood and other benefits. Reason why we don't want to it may not be true, if we twist oauth2 in an unnatural way and people think its oauth2 and its not then we won't, use a new custom end point and semantics, still using JWTs, but custom as cant make it fit.
- Kurt: possible to conform to a spec without being interoperable, because of extensions to or corner areas of a spec, something may be considered legal but doesn't mean that off the shelf software is going to understand it or operate in that way. When we talk about differences between the two not enough to say whether something conforms to a spec but that the interoperablility?
-
- Andrew: that would go to Craig's point if you are going to end up getting into that realm of maybe matching spec but no other interoperable clients, have gone away from OAuth enough that let's not confuse people by saying it's like OAuth. If we have a divergence then declare this. Like if no existing OAuth clients would work with it. We need to decide if we can go OAuth the whole way and have a vague sense that

we have an oauth and would be interoperable with general OAuth clients out there. Or we decide to diverge off oauth flows off at this point and not even pretend that it looks like oauth.

- Max: let's frame what is at stake going either way. Don't see the primary issue of this particular subproblem as interoperability, whether or not it's interoperable depends on what you do with the token, anything that can do oauth can get a token, it depends what you do with it later. What is at stake is not so much interoperability for this one subproblem, it's consistency with what you do next and what you do next is this is the big issue of interoperability. If a priori do oauth down this process, Stick with oauth now to make consistent API but if don't then fork it off now.
- Kurt: once go through your exchange, created an adaptor which is no longer same thing. What we are talking about as GA4GH is not what you do privately but what you do publicly, how interoperability is defined between organisations that use GA4GH standards. Throwing an adaptor into the situation, it changes the model.
- Craig: couple of things come up but think that let's see what does fit in OAuth 2.
- Susan: an interesting point made a second ago, mention of basically the infrastructure already exists around OAuth and existing clients. New people coming to this, is there an issue of support and tooling that's available at their institutes? Is there an adoptability argument here for other institutes?
- Craig: put this as a requirement, how much do we want interoperability in the middle nodes? Issue token has visas in it invented at GA4GH, entity inspecting these things is not going to be an off the shelf thing, but knows what to do with GA4GH, but in the middle passing it around or expecting log ins and slows, that is an area where we may want to stick closer to interoperability.

- Andrew: identity providers you want to be oauth, don't want to be minting custom GA4GH google log in end points or anything like that. Lots of existing identity provider infrastructure, We don't want to be touching that in a custom GA4GH way.
- Kurt: I put an appendix in document, that describes what NIH's flow and conception of this flow is, starting point entirely 100% is OIDC and OAuth up until point you get the passport. So would have the support for those authentication mechanisms.
- Craig: we still want that compatibility, which is Andrew's point.

- Craig: features we need to explore in here. Notion that these custom passport tokens aren't where the user logs in, it's after that. And oauth2 has some mechanisms too, could max talk about offline tokens and other stuff that is available as part of the spec. believe you could still create the JWT token to have inside of it whatever you would like. User info end point to grab the extra information.

- Max: as a general point, oauth has a lot of different cut points where you could choose to extend it here. Functionally not a huge difference to a new end point and an oauth end point with a slight extension. Value is on what conceptually binds these

APIs together, what does this have to fit in with? Is there value in making it similar to other parts of the system it interacts with.

- There are a few points you could extend to: one at token end point takes a grant type parameter and you could make custom grant types. That's the first and most broad cut point where we could make a new GA4GH standard. To get a passport with embedded visas, you can use this special GA4GH passport grant type, usg existing end point, we should use similar kinds of conventions that other grant types do, like client request, but we will have the flexibility to give it the semantics we want. That's one option.

- Second cut point: already RFC for token exchange, already defines this in a similar way to what we would want to do but gives you a token type parameter that can be extended. So you can say this is the type of token I can have, and this is the type of token that I want. I am passing in an access token and I want to get back a passport token. We could have a GA4GH standard saying this is the canonical token type of a passport with embedded visas.

- Similar to how OIDC extended OAuth to add an identity token, they hijacked the authorisation code grant type for that, did not invent a new grant type. Required a new scope. We already require a new scope to do a passport flow, we could easily all in one flow with token response when you log in the first time, we could include an additional JWT to what we already have that contains those embedded visas. This third option is the simplest one but it critically wont set us up to solve the problem in the future of downscoping, so that's a trade off.

- Other two are good options, if think that OAuth is going to appear more downstream. Then we can keep a consistent API when we design this.

- Craig: question on some of these things, what can we and can we not get away with. Token exchange that can be extended and we wanted to do downscoping. Can we use them together?

- Had a question on some of these things, trying to pressure test what we can get away with. Number 2 with token exchange, token type that can be extended and do downscoping, can we use them together? Can we pass in resource exchange?

- Max: there is no explicit RFC that tells you how ot use them together, but in DNAstack we do use them together. Orthogonal, can add resource indicators to any existing OAuth request for token end point. GA4GH could have important role in making the RFC more specific for passports, what does a URI for a visa look like? Give our specific GA4GH interpretation.

- Andrew: this would be a token exchange send in a token type, passport and token back type of passport and put in a downscope to this.

- Max: yes that's how I would imagine it.

- Andrew: could use same to exchange a big passport for a big access token. Just changing the token types.

- Max: if want to go that route up to us to say within paradigm of a passport broker, which of these make sense and when are we allowed to do these things.

- Craig: use some resource indicator RFCs to look like this (see spec). with authorise use token exchange or other mechanisms. Our profile would tie them together for how we use them for GA4GH

- Max: is this just meant to model the token exchange request?

- Craig: not different end point here.

- Max: if went token exchange route, nothing would change with authorise end point in passport flow today. One extra step at the end where you would take the access token from passport flow and send a request to token end point. I have a token of tyope access token, I want a passport token and here is my client ID and secret in the header. Get back a request payload that has new embedded visas token in it. Beginning of flow does not change at all, only what you do next.

- Craig: sounds most similar to custom passport token proposal. After have original access token from other flow, go grab this other one and also use other token that you get, passport token, can use that to exchange again and subset what you have again. But sounds like could be done with token exchange, most similar to what the custom proposal is.

- Kurt: describing broker mechanism

- Craig: in custom passport side of things, refresh token? What a refresh token would look like. Don't want to issue refresh

- Kurt: no that's not true. Its not that we don't like refresh tokens, way that refresh tokens work that when you contact in oauth2, authorisation server you can get back an access token and a refresh token, considered secret and held only by you, access token goes into public, only way refresh token works and reason it is exists is as a revocation mechanism, forces you to back to server periodically to get an up to date access token so server has opportunity to revoke your access. Only works if refresh token kept private. As you pass access token down the chain you cannot also pass your refresh token. Instead, in Max's model, the intermediate authorisation servers that are not the original authority, issue their own refresh token that has no guarantee of having any relationship to the refresh/revocation mechanism of the original authority. If there is an intermediate server that becomes authoritative voice or statement behind who can access what, it cuts the actual original authority out of the picture. Original authority could try to revoke access where the intermediary, this is the problem. So since refresh tokens can't be passed down, the OAuth flow does not support that. Introduce the chain of trust and therefore drop original authority.

- Craig: each time you go down the chain you get more refresh and access tokens. So check with max on token exchange. Can we do token exchange even on our own custom token type? You wouldn't necessarily need to get a refresh token to get another one of these things.

- Max: refresh token has its own grant type, special grant type. To exchange that for new access token. Token exchange generalisation of that.

- Susan: if there is a chain and originator wants to revoke access, and tokens passed along how is that controlled? Can restrictions be put on part way down the chain?

- Max: passing down the chain gets to priority before when does the clearing house happen. In this case for this one end point, follows standard OAuth rules, do token exchange, get JWT, how do you deal with revocation.

- Andrew: refreshed token, genomics has some unique use cases of passing down tokens that go to another system that needs them for 12 hours to run some compute job, it is outside normal mechanisms for refresh tokens. What are others doing in OAuth world for long running compute?

- Kurt: not unique to genomics but genomics one of leading cases. Someone ask me once in NIH what are other agencies doing that we could look toward. Answer is they are looking towards us as we are handling big data. We are a forerunner.

- Craig: close with token exchange piece. No separate refresh token required, but circle back next time with Kurt. Go back to source of authority and check if it has been revoked. Address this again. Does this capture the capability that we could specific (Max?) and with Kurt does this capture some of the requirements?

- Max: scope looks good to me
- Kurt: from our standpoint trying to reduce number of token exchanges as opposed to increase. Interested in seeing a token from it's source all the way to the clearing house. Nature of flow outlined by Kurt at end of document. Read to have more of a view (Appendix)

Some chat comments:

Kurt: The OAuth approach of chain of trust creates multiple refresh tokens that are not guaranteed to reflect the authority of the original. It reflects the judgement of the intermediate servers, whereas the design of GA4GH passports is for the refresh question to go all the way back to origin (indeed, this is how revocation works).

Kurt: Using an OAuth flow means that the original refresh token cannot be passed down the flow, and this severs the path back to the origin. These are real-world impediments of the OAuth design to the GA4GH use cases.

The OAuth issues revolve around refresh mechanism that is not amenable to a multi-stage workflow (WES?), because the refresh token cannot be passed down the flow without destroying revocation. The way that OAuth deals with chained workflows is by using chained authorities, and that destroys original authority, including that of the refresh token.

2021-07-23: Second AAI/Passport hackathon concentrated group meeting

Chair: Craig Voisin
Attendees - Name (Affiliation): Craig Voisin (Google), Kurt Rodarmer (NIH), David Bernick (Broad), Max Barkley (DNAstack), Susan Fairley (GA4GH), Alice Mann (GA4GH), Jeremy Adams (GA4GH)

	Actions Arising	Assigned To	Deadline
1	Alice to share notes	Alice	
2	Plan some requirement gathering strategies in a	Craig/Alice/Susan	

	GA4GH centralised way		
3	Arrange next meeting +Mikael Linden, investigate if Andrew Patterson could join	Alice	Next two weeks
4	Review PRs and feedback to Craig if any changes needed • AAI PR: https://github.com/ga4gh/data-security/pull/46 • Passport PR: https://github.com/ga4gh-duri/ga4gh-duri.github.io/pull/55	All	

	Agenda Item	Person/Time
1.	See summary below for discussion points	
2.	Review: https://github.com/ga4gh/data-security/pull/47	
3.		
4.		
5.		
6.		
7		

Minutes/Notes

Status at the start of this meeting:

- We hit a milestone: terminology of Passport and Visa have now been patched in v1.0 of the AAI profile as well as the Passport spec.
- AAI PR: https://github.com/ga4gh/data-security/pull/46
- Passport PR: https://github.com/ga4gh-duri/ga4gh-duri.github.io/pull/55

We had 2 reviewers (David Bernick and Kurt Rodarmer) for those PRs → rest of team review and let Craig know if need any corrections however if you think anything is amiss, please let me know so we can make corrections.

Next step:

- Craig has prepared AAI and Passport PRs for v1.2 to support self-contained passports. Goal will be to have a draft PR as input to the next round of discussions.
  - https://github.com/ga4gh/data-security/pull/47
- To land v1.2, we will need to include details such as what endpoint to hit on the Passport Broker, however there are still issues to resolve there in terms of what endpoint to use and what properties it should have (e.g. is an OAuth2 token endpoint or a new endpoint.
- Note that we will attempt to land the Passport/AAI v1.2 changes even before the DRS API discussions resolve such that the flows with Brokers is well understood. Therefore I'm treating DRS endpoint as out of scope, but clearly related so keep in mind when reviewing.

I'm requesting the following action items for you to do (except Tommi and Mikael Linden on extended holiday):

- Please review and comment on the "Client Requests Downscoped, Self-Contained Passport" draft doc that will be used as a rough starting point for creating the PR.
- In particular, weigh in by adding or commenting on the "Features of a custom passport token endpoint vs. using OAuth" subsection such that we have as much input as possible to frame the next hackathon session.

Meeting Discussion

We aim to get down to requirements, all agree on these and what's the best way to solve these requirements.

Helpful for Susan to represent the community at large as best as you can, the people on this call don't represent everyone.
Looking at this table, there is the detail we need to talk though. Craig tried to cluster actions into 3 major topics that we need to decide on:

1. Mint new tokens while retaining Original Authority
2. Building on existing standards and embracing a wide set of tools/developers
3. How much support for having multiple ways to get things done?

We don't need to make individual decisions but can discuss the requirements around these 3. How interoperable are these things?

We mostly discussed point 1 and the issue of retaining Original Authority

- Minting new tokens while retaining original authority, key features we got out of the passport spec is the idea that we retain original authority by treating each visa as a

separate string, wherever it goes, original signature maintained as it goes through multiple systems. Make sure we understand how we want to retain original authority.

- Kurt: essentially the authoritative source either the resource owner or steward in case of NIH within NIH, will create a token that the journey starts there, token bears authorisations signed by authoritative party, those authorisations and token are preserved intact all way to end where token is exercised at the point of access to data, end point is the clearing house. Token clearing house that receives the token, validates it, does all integrity checks operates in the interests of the data host and data owner. Everything in between, the token is resilient to being handled. Meant to make that journey and handled by the clearing house.
- Subissues, what content do you need to have to authorise at the clearing house. Make sure this is maintained.
- Original authority in workflows w/o need for refreshed tokens is a subgoal, is this a good subgoal? Useful feature of approach?
- Lastly, introducing downstream tokens that do the work, another thing we are looking towards not necessarily done in v1.2. looking ahead at what is coming next, in custom approach maybe another custom token 2 and then oauth approach as to what you would do to do this kind of thing.
- Issue is OAuth2 loses original authority, when it was designed for a particular set of conditions but when it is extended to GA4GH conditions, it drops any trace of original issuing authority from the chain and uses a chain of trust instead (trust me with this token that NIH have authorised data rather than maintaining the token issued by NIH originally). Chain of trust is legally unacceptable as the tokens become substitutes for legal agreements.
- Kurt fought hard in v1.0 to get the original authority to be preserved, this was the origin of the embedded visa. By definition therefore it is custom, not OAuth.
- The idea behind the token retaining original authority has taken us away from the standard OAuth. OAuth uses an exchange mechanism, which means dropping original authority. It doesnt even use a data steward, just considers the user to have signed off.
- David B: what about having multiple ways? Some users want to make sure original authority is maintained but some way within the stacks that we exist, it’s fine to replace the tokens with something else in our visas. So can we have two different ways of getting things done, not millions?
- We do already have a middle party that grants access to data using NIH delegated authority, Tera.
- Kurt: but the policy will be amended so authority coming more clearl from NIH, this was managing authority and data storage as there was no other mechanism for handling otherwise. Now authority is centralised.
- Post clearing house, no passports or visas. Expected to be oauth tokens but aren’t supposed to reenter GA4GH ecosystem as in competition of passports as tokens of authority.
- Mechanism that tera uses to give access to data is to give a token after processed the passport.

- Susan: data stored on tera. Some sort of question with trust. Handed a copy of data over to tera, control of who accesses rests with tera and tera infrastructure. As soon as you hand over data, control has been handed over?
- Kurt: assumption above, may be true in some cases, not sure if necessarily true. To susan's point, there is an element of trust for people in security, the root cause of all breaches is trust. Trust extended unnecessarily is nothing to do with trustworthiness. When design security system only extend trust when no other recourse, principle of leased authority. As move to systems with least trust (zero trust/reduced trust), as we move towards zero trust it means we have more validation at every steps, even at steps we didn't have them before.
- An approved clearing house is part of that. Clearing house needs to see tokens coming in from authoritative source.
- Craig: then there's the issue of downscoping, unlike real humans we don't have one passport, but multiple different passport tokens. Passport can be scoped for a particular use case. So don't want to lose original authority when we get a new one, or expand to maintaining original authority all the way through.
- Kurt: intention is passport all the way to DRS server, DRS hands to clearing house because this is highly trusted operation, clearing house should be operating in same security boundary as DRS. Token ends it journey at CH, internal to data host system or resource server system to stay within OAuth, use signed urls not because good tech but what we have now.
- Third issue, how many ways we preserve original authority to clearing house.
- Custom passport: passport is bearer token, downscoped tokens must retain original authority. Passport downscoped to some visas.
- Oauth approach: take original passport and use oauth 2 endpoints and get oauth2 type token that had just 3 visas in it. Signature to say who allowed this and who has authority. Checking passports, does string say you are allowed to read email but who says you are allowed to read email and verify that's the right entity. Oauth2 you could still get strings that are longer and these strings represent visas. Can you use oauth2 to keep visas somehow or are they self contained?
- Kurt: think you can but as soon as you do you have custom passport token, exactly what it is, JWT with all same stuff but has tokens inside, instead of having claim that says max says susan can access data that belongs to ebi, instead says max gives you a token which has a token inside that EBI says yes susan is authorised. The resource server believes EBI not believes max.
- Craig: retain visas so they are still there, custom claims.
- No can't still use oauth2, involves refreshed tokens and token exchange which is unless retaining internal visas is problematic and whole point of max's argument that they fit inside a small header and not carry those visas and otherwise usable by existing tools. Oauth2 + something, not the same. No existing oauth RFC that handles this.
- Max: one endpoint where you can give self-contained passport that have many visas and get back one so has fewer visas. Is this a new end point? Or most of end point most of the way there , with token exchange oauth2 nearly there. Or create token. Otherwise 2 approaches same. Do we define whole API or token exchange +

- Kurt: but what is achieved? You've taken a regular API and modified it in a way that nobody does. Clearer if we introduced a new end point and new end points are not a big deal.
- Which endpoint downscopes the token is where the value of sticking with oauth is lowest in value. a new endpoint isn't a big deal, although i think API design cohesion is a consideration.
- I find new endpoints bring clarity to function. It's their purpose in fact. One server, different endpoints.
- Can we do an open one week, feedback, we think these are the requirements?

We've mainly talked about point 1 but there are also points 2 and 3

1. Building on existing standards: can we build on existing standards and use them as much as possible or is there a reason why we chose not to use it. High level requirement, if a standard can be used, requirement is to use it and if it doesn't mean our wider goals then we don't use that requirement. Cost of developing implementations, existing tools means they hopefully lower the cost.

2. Having multiple options, how much support do we want to build into our specs. Even worse if we choose on option or the other, even worse if do all 3. More options offer, less interoperability as people will chose to implement subset of the spec. try to make as few ways of doing things as possible, if do do multiple ways have some strict requirements of what it would take to have multiple ways of doing it.

Clear take home message from this meeting:

We have a requirement: we need some way to take a token in and be able to spit out a token that has less stuff. The rest of the discussion is based around which path you could take to get there.

An action therefore is to get the requirements of what the community needs, what are the actual requirements as opposed to getting different standards.

2021-06-29: First AAI/Passport hackathon meeting

Chair: Craig Voisin
Attendees - Name (Affiliation): (wider aai group)

	Actions Arising	Assigned To	Deadline
1	Land 1.0 patch/PR next month		

2	Work on landing next v1.2		
3	Reach out to David B/Craig if want to be explicitly tagged as reviewer on PRs.	All	
4			

	Agenda Item	Person/Time
1.		
2.		
3.		
4.		
5.		
6.		
7		

Minutes/Notes

- AAI will follow whatever passports decides is their mechanism of transport, AAI will probably collapse into passports.
- David: suggestions hopefully will accelerate that. Whatever passports decides, AAI will make happen.

Passports/AAI terminology updates

- v1.0 specs AAI and Passports, making sure terminology makes sense.
- Not trying to make breaking changes, supposed to be a patch if works successfully.
- 1.0 V Updates for AAI spec and Passports.
- Draft v1.2 AAI terminology, influenced by work Kurt has been doing, self-contained tokens, as an extension to the API, a new point version, hopefully backwards compatible, how we would extend in this direction.

Corresponding GA4GH bearer token added:

- Thinking is to avoid access token may want to reuse it later, might imply using a certain OAuth flow that we are not committed to at this moment.

- Called it bearer token, do attach it to a request and so we are just saying this is the thing that you can use as part of the AAI spec where you get a token that is used as part of returning from the OAuth end point, already in the spec.
- Open question: OAuth
- But have bearer token that can come back and use at various end points

- Reword some of this embedded token issuer, talking about the fact that we have the embedded tokens, spec should be highly aligned maybe, should it be closer to a visa? for the passport spec

- David B: GA4GH bearer token in 1.0 is an access token, it's not the stuff returned from user info even if stuff is packaged as JWT. Later in 1.2 they'll be a self contained token, returned stuff from user info or other end point that is a JWT that can be passed down to DRS as have the visa.

- Craig: yes so we need some text changes, user info is an end-point, so this isn't explicit enough. It is an access token in 1.0, in the sense of a traditional access token. But in v1.0 we could be explicit and say it is an access token. Okay agreed and added self contained token.

- David: be clear that this is the OAuth access token that does not contain visas, very explicitly, make this clear. It is used to access the end points including slash user info.

- GA4GH claim: where visas can live. It does not contain GA4GH claims, contains the scope but not the claims. The user info contains the claims.

- Kurt: at this point in the flow, GA4GH has not yet entered the picture, everything is OIDC

- Craig: true except for specific scope, it's a GA4GH scope.

- Craig: We need a better name for visa issuers. We have the embedded JWTs and embedded JWT issuers.

- David: Why don't we introduce passports and visas even as ancillary. We are joining these things. It's only an illusion that these are separate. So we could add that these are used as passport visas and then link to the passports document.

- Kurt: what we are talking about in passports is completely separate from the semantics of the claims. Whole idea of DURI and DUO, DACs and semantics of the claims, not part of this discussion. I think we are talking AAI, merging great from my point of view. Call it passports.

- Craig: V1.2 make it the official name of visa issuer and visa. Put the old name in case anyone has to reference the old name.

- https://docs.google.com/document/d/1ISRIJRFSIB8EMww_yOY6hWkT6O7jDABdmxegsmFBD24/edit#heading=h.kavlm75yur0u

- David: Tom is a new employee at Broad who works in David's team.

- Craig: passport corresponding changes, leave some things to AAI spec as to how tokens work, this is more of a format.

- Vivian: like high level and will transfer to the policy community, more accessible.

- Craig: passport, includes anything you need to get your permissions. Cover of the passport, token itself, separate from the pages of the visas (technical construct).

- Do we need to have passport bearer token as a separate thing? Should the visa be called a passport token in the AAI spec.

- Max: I like that everything has a passport adjective token, passport bearer token, passport container token. Passport bearer token does not have the claim in. v1.2 passport something token rather than self-contained passport. E.g. passport self-contained token in v1.2 passport terminology.

- Jump into v1.2 and try to align v1.0 definitions.

- Kurt: GA4GH introduces new passport and visa.

- Craig: e.g. bearer passport not passport bearer token?

- Kurt: Passport by definition is a bearer token.

- Craig: in the past, the way we talk about it uses passport to represent everything.

- Kurt: we hear this subject at NIH a lot, worried about protecting past mistakes and the general feeling I hear at NIH, let's stop doing that and do it right and go forward. Passport should be a thing not a brand.

- Use passport by itself: means entire permissions everything, throughout specs generally not use it by itself, bearer passport for example.

- Kurt: all bearer means is that it hasn't been assigned to an identity, it's a class not a thing. Bearer token doesn't tell me anything other than makes me suspect that there's an assigned type of passport which there is not.

- Craig: go back to 1.0, OAuth access token. Not just any token for Passport bearer token, it's the passport one, with the separation between AAI and Passports. Has the passport scope in it. So what do we call bearer passport?

- Kurt: Passport-Scoped Access Token - OIDC token like any other, fact that it contains one scope compared to a different scope, that's what OIDC access tokens do. Same mechanism of getting you that token does not have to contain passport.

- Andrew Patterson: but why we talk about it here is that you need it to get a passport.

- Craig: okay good but what shall we call GA4GH bearer token if we call it passport-scoped token?

- Max: what about AAI-scoped access token?

- Mikael: but there is no AAI scope

- Craig: true what about a GA4GH scoped access token? Or is this the same problem, don't have GA4GH as a scope

- Mikael: GA4GH better than AAI scoped.

- Craig: we don't talk about scope yet in this stage so is this ok? OAuth2 access token, enough? Or do we need an extra qualifier?

- Kurt: The GA4GH Bearer Token was a vague name wanting to dance around the word Passport, from what I can see. If we are merging, I don't think it will cause confusion but will clarify language for people. We should go forward with this.

- Michele: people don't know where to look when they go into this, maybe make the relationship a bit more clearer.

- Craig: could we just make this defined passport scoped access token here? And literally have the scope name in AAI spec already and move it down, make be clearer than wrap it.

- David: already said we are going to merge them so I don't think we need to pretend that we aren't. by v1.3 they will be merged. V1.2 is the last separate AAI spec

- David B: whatever gets us to federated access!

- Mikael: align definitions in this version the align the texts and documents too.

- Craig: we could do even more in this direction, if we are doing that could define user info visas as user info is discussed in AAI spec so could move a bunch of definitions down.

- David: can do offline that we are going to merge definitions and in agreement

- Craig: jump to v1.2

- David: letting NIH define 1.1 by means of having done it, doing it is standard enough so are we going to skip 1.1 because of this or just going to accept it and move on?

- Kurt: sure we will publish our 1.1, 1.1. like many spec proposals that you just go onto 1.2 to create the spec.

- Craig: AAI 1.2 terminology, self contained token. We are going to use it to pass down to DRS so should we use it as bearer token? Just call it a passport token or something, JWT containing the claims that can be used to authorise downstream services.

- Mikael: no scopes in self contained passports

- David: no in visas, self contained passport just a bundle of visas.

- Mikael: grant permission to call an API, it doesn't do it by using a self contained passport as it doesn't have scope and scope is the regular way to do this
- Kurt: scopes are sort of the way and poor mechanism to call visas, visas are much more specific. Mechanism coming forward for downscoping the passports and user will have the ability to downscope. In today's scopes, single scope defined for visas, GA4GH v1, gets you all visas, not scoping as such as isn't much fo a scope. Still available tp user as user gets to the passport through OIDC goes through log in and consent for the scopes to the OIDC access token and only based on that access token can you get the passport or not depending on what the user has consented.

- Mikael: but if want to authorise API call based on scope that auth server to client, do we completely abandon that OAuth approach?

- David: still keeping OAuth in OIDC flow to get passport but once have passport we divert from the flow

- Kurt: visa is specific and scope is not

- Max: scope issued as a boundary for things you can do not specific data.

- Kurt: that can be expressed as a visa as well but in general visas are more specific than vague sort of class of things.

- Max: WES server, don't know any use cases in the short time where you'd use a passport to get….. want to use self contained passport to get access to the data.

- Craig: passport term down in AAI spec so delete one at passport level. What are the next level edits we want to make to this bloc?

- Max: need to make recommendation as to how you include this token in the body for certain popular content types, otherwise leaving it to very spec, DRS, Data connect to do that

- David: that's good but can we dictate that without getting their buy in first?

- Max: add clause, you "should" use this attribute, let sit for v1.2 for v1.3 it becomes a must and we give the attribute a name space name that it's unlikely to collide.

- David: anything else we need to hit in next couple of mins

- Craig: refreshed tokens later. Check in Mikael, anything he would like to see/direction we can capture and keep ELIXIR in mind?

- Mikael: wider questions, what is the value of this? Self contained passport if visa tokens are the regular way to do refreshing. Make sure visas in self contained passport belongs to the same user.

- Craig: v1.0 continue to keep backwards compatibility. Land spec changes to allow for self-contained passports but probably not require them so can continue to use backwards compatibility.

- Mikael: may adopt self contained passport when it shows its full power.

- Kurt: For NEW protocols, we can recommend POST as being a safer bet that works in all cases. For legacy systems, we need to have the "indirect" passport, which is small like OAuth2 access tokens.

- Reach out to David B/Craig if want to be explicitly tagged as reviewer on PRs.