

## INTRODUCTION :

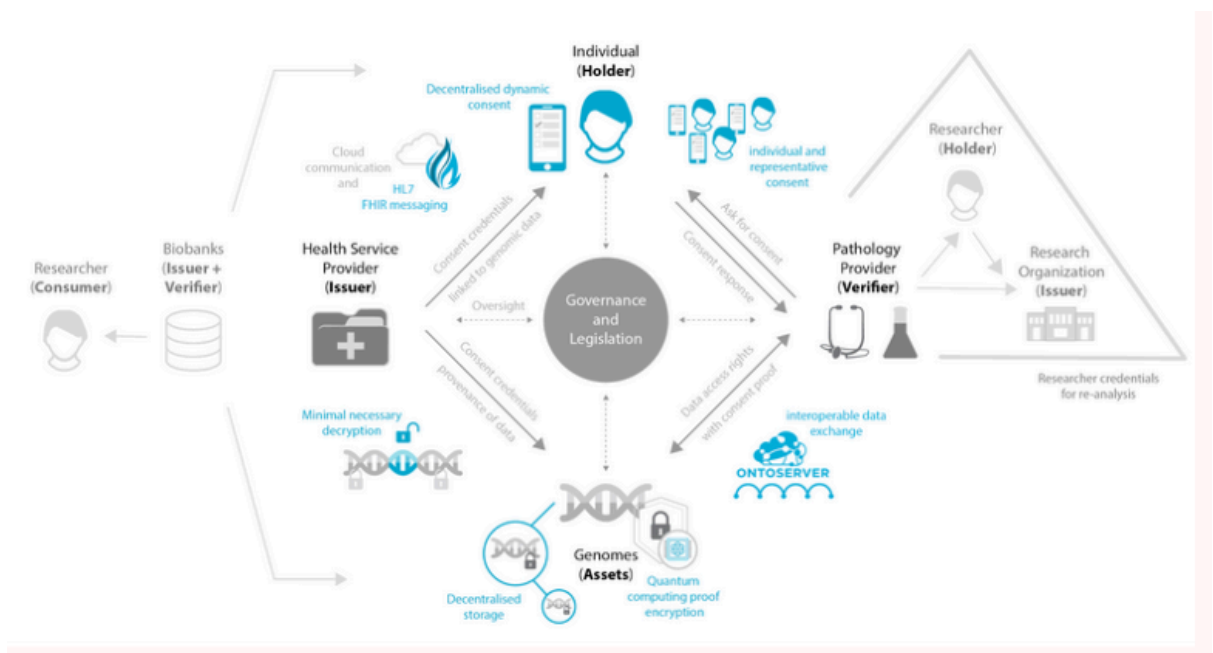
- This work is part of the Gene Guardian project, previously named Dynamic Consent Protocol, which suggests a perfect framework that puts the individual in the middle of the decision making.

As we know the use of **genomic information** in medical treatments and disease risk management must **balance** personal and societal benefits against risks to individuals contributing their data.

**Trust** in the professional and ethical handling of genomic data is crucial for participants to consent to the use of their data.

- in health care service :
  - when I'm sick, i will go to the hospital. The clinician tasks the **healthcare service provider** (the issuer) to initiate a genomic test after obtaining patient consent
  - The **pathology provider** (the verifier) checks the identity to create the genomic data
  - **After verification** by the issuer, they send it to the **patient**

figure 1 : diagram of the medical application



- so, my role is : Create a **form** using **FHIR resources (& adapting it to our needs)**

This form is a **mixture** of forms of **hospitals** and a form already created by the team and **later on** interacting with a **server**.

- What is HL7 FHIR ?

It was created by the standards development **organization Health Level 7 (HL7)**. It was designed to enable **health data** to be quickly and **efficiently exchanged**.

FHIR is a set of modular components called "Resources." These fields include basic elements like identifiers, metadata, and attributes relevant to the type of resource.

example of a resource :

Structure	UML	XML	JSON	Turtle	R4 Diff	All
<b>Structure</b>						
Name	Flags	Card.	Type	Description & Constraints		
Location	TU		DomainResource	Details and position information		
identifier	Σ	0..*	Identifier	Elements defined in Ancestors: Unique code or number identifying the location		
status	?! Σ	0..1	code	active   suspended   inactive Binding: Location Status (Required)		
operationalStatus	Σ	0..1	Coding	The operational status of the location Binding: hl7VS-bedStatus (Preferred)		
name	Σ	0..1	string	Name of the location as used by the location		
alias		0..*	string	A list of alternate names that the location uses		
description	Σ	0..1	markdown	Additional details about the location		
mode	Σ	0..1	code	Instance   kind Binding: Location Mode (Required)		
type	Σ	0..*	CodeableConcept	Type of function performed Binding: ServiceDeliveryLocationType		
contact		0..*	ExtendedContactDetail	Official contact details for the location		

- HL7 FHIR has evolved **through four releases** since its initial presentation in May 2012. It has grown from a true draft standard with 49 **Resources** to its current **145**
- In that time, the standard has improved and changed to **meet the needs of the health information technology community**

## THE WORK :

### First step :

- Since we are talking about a form, it means we have questions, and the FHIR resource we use is the Questionnaire resource, which has different attributes: id, url, the name of the form... and the most important, the items.
- The items are structured in this way: linkID which contains the ID of the question, which we can use later, the text attribute which contains the question itself, and the type of question : boolean (where we answer true or false), a group of questions (which in turn give sub-items), and choice: where you make a choice between the given proposals..
- We can add extensions : like a duo code that may correspond to each question. To do that, you must absolutely attach an url (for the type of extension added every time).

Second step : Our questionnaire is in a JSON file. To run it, I've created a python script that contains different functions.

- indispensable function :

```
def print_question(question)
```

- It displays the questions of the questionnaire with an input for the patient's entry, and it encompasses all types of questions.

```
def save_to_file(data, filename)
```

- it opens the file (our data) and saves it on the laptop in JSON format.

```
def get_duo_code(item)
```

- It returns the duo code according to the extension URL found in the item, if it corresponds to the duo code.

```
def send_to_hapi_server(resource, resource_type)
```

- It allows us to send the resource we've created to the HAPI FHIR server to test the proper creation of this resource.

- If it has been well sent, we have a code that is displayed which is 201, with the ID of this resource in FHIR (which we will adjust in the creation of our resources to ensure the proper reference between them, because they are created and sent to the FHIR server).

- functions to create the resources needed :

“ We create as many as we need and link everything to our resource of interest, which is **consent**. And to link the created resources with the consent resource, we need to take into account the IDs, and the IDs currently used are those given by the server ( returned by the function `send_to_hapi_server` ).... ”

```
def create_questionnaire_response(questionnaire, responses,
questionnaire_response_id)
```

- The first resource we need is the one that takes into account the responses to the questionnaire: QuestionnaireResponse. And which has as attributes: the id, the questionnaire with its url or ID (if sent to the server) , the status, and the items which contain the answers.
- By creating an `add_item` function that each item takes the answers, taking count of its type and adding it to the item attribute with the right linkID, and also adds the extensions (`duo_code`) if present.

```
def create_practitioner_resource(practitioner_id,
practitioner_name)
```

- it only contains a few information such as the name and the ID.

```
def create_patient_resource(patient_email, unique_id,
questionnaire_response, questionnaire_response_id,
practitioner_id, birth_date, practitioner_name):
```

- We also have the Patient resource that takes as parameters the patient's email, the id, the responses of the questionnaire, the id of the questionnaire responses, the id of the practitioner and the name, and the date of birth.

- We add additional questions such as the phone number and create our resource that contains as attributes: the id, first name last name, telecom, gender, the practitioner (and we link it to the ID), and we add an extension (which must have a mandatory URL which is the one corresponding to the questionnaire response) with the ID of the latter in the reference.

```
def create_consent_resource(unique_id,
server_practitioner_id, responses, duo_codes)
```

- We create the consent resource that takes as parameters : the ID, the server\_practitioner\_id, responses which are the questionnaire responses and the duo\_codes.
- The consent resource contains the resources we've already created before, and we link it to our consent as well.
- In our consent resource: we have the ID, the category of the consent (which is Research), the dateTime, the name of the organization, the source of reference: we can use the responses of the questionnaire, and we also have the provision attribute which itself can contain other attributes.
- for the provision.consent : i will detail two methods
  - 1st method** : I thought about creating two functions: create\_consent\_provision and integrate\_provision.
- For the **first function create\_consent\_provision** :

```
def create_consent_provision(response, duo_mapping,
server_practitioner_id)
```

- It takes as parameters the response, the duo mapping, the server ID  
We create an empty list consent\_provision. And we go through our responses and in each linkID where we have duo mapping, we create the extension, and we add the response of the patient. Then, we append all this to our consent\_provision list.
- We therefore obtain data that contains an 'extension' that has the duo code followed by the corresponding response (depending on the duo mapping already done previously).

- The 2nd function :

```
def integrate_provision(consent,  
questionnaire_response, duo_mapping,  
server_practitioner_id)
```

- This function will allow us to integrate this data into our consent resource's provision attribute. It takes as parameters the consent resource, the questionnaire responses, the duo mapping, the ID)

### 2nd method :

```
def create_consent_resource_withprovision(unique_id,  
server_practitioner_id, questionnaire_response, duo_mapping )
```

- I modify the function create\_consent\_resource by adding a boucle 'for' that iterates and items of duo\_codes and put the code in purpose attribute, and add the 'type' depending on the value if it's true or false, it will take 'permit' or 'deny' answer.

### Third step : create our main\_function()

```
def main()
```

- The script reads a complete questionnaire from a JSON file and collects user responses.
- It maps specific DUO codes to different research purposes using the 'duo\_mapping' dictionary.
- The user's responses are used to generate several FHIR resources: 'QuestionnaireResponse', 'Practitioner', 'Patient', and 'Consent'.
- The 'QuestionnaireResponse' and 'Practitioner' resources are created, saved locally, and sent to a HAPI FHIR server.
- The server-assigned 'Practitioner' ID is retrieved and used to create the 'Patient' resource.
- A 'Consent' resource is also created, incorporating DUO codes to permit or deny specific research purposes based on the patient's responses.

- The `Consent` resource is then completed with provisions, saved locally, and sent to the HAPI server.
- The script includes error handling for file reading, JSON decoding, and the creation/sending of resources.

## Conclusion :

- example of QuestionnaireResponse resource : you will find all the informations needed : the items with the answers and the extensions that contain the duo-codes.

---

```
{
  "resourceType": "QuestionnaireResponse",
  "id": "44937925",
  "meta": {
    "versionId": "1",
    "lastUpdated": "2024-08-26T01:05:03.466+00:00",
    "source": "#AGGPNQhdMYCFQQKD"
  },
  "questionnaire": "http://example.org/Questionnaire/genomic-testing-details",
  "status": "completed",
  "item": [
    {
      "linkId": "1",
      "text": "It is my choice for my child/person under my care to have genomic testing.",
      "answer": [
        {
          "valueBoolean": true
        }
      ]
    },
    {
      "linkId": "2",
      "text": "[Parent/Guardian Name]",
      "item": [
        {
          "linkId": "2.1",
          "text": "Family Name",
          "answer": [
            {
              "valueString": "jdfjgfdg"
            }
          ]
        },
        {
          "linkId": "2.2",
          "text": "Given Name",
          "answer": [
            {
              "valueString": "gjfgjfd"
            }
          ]
        }
      ]
    },
    {
      "linkId": "3",
      "text": "I understand that my child's/ the person under my care's DNA will be tested by:",
      "answer": [
        {
          "valueString": "Genome"
        }
      ]
    },
    {
      "linkId": "4",
      "text": "I provide consent to share my child's / person under my care's sample, genomic data, and related health information for :",
      "item": [
        {
          "extension": [
            {
              "url": "http://example.org/fhir/StructureDefinition/duo-code",
              "valueCoding": {
                "code": "GRU"
              }
            }
          ],
          "linkId": "4.1",
          "text": "General research use",
          "answer": [
            {
              "valueBoolean": true
            }
          ]
        },
        {
          "extension": [
            {
              "url": "http://example.org/fhir/StructureDefinition/duo-code",
              "valueCoding": {
                "code": "HMB"
              }
            }
          ],
          "linkId": "4.2",
          "text": "Health/medical/biomedical research",
          "answer": [
            {
              "valueBoolean": true
            }
          ]
        },
        {
          "extension": [
            {
              "url": "http://example.org/fhir/StructureDefinition/duo-code",
              "valueCoding": {
                "code": "NCTRL"
              }
            }
          ],
          "linkId": "4.3",
          "text": "For research involving the following area",
          "answer": [
            {
              "valueString": "cardiak"
            }
          ]
        }
      ]
    }
  ]
}
```

- example of Consent Resource :

FHIR server : one provision by consent resource.

### Payload

```
{
  "resourceType": "Consent",
  "id": "44938005",
  "meta": {
    "versionId": "1",
    "lastUpdated": "2024-08-26T04:30:52.673+00:00",
    "source": "#snUanqOMx1fwtxXU"
  },
  "status": "active",
  "scope": {
    "coding": [ {
      "system": "http://terminology.hl7.org/CodeSystem/consentscope",
      "code": "research",
      "display": "Research"
    } ]
  },
  "category": [ {
    "coding": [ {
      "system": "http://terminology.hl7.org/CodeSystem/consentcategorycodes",
      "code": "dna",
      "display": "DNA analysis"
    } ]
  } ],
  "patient": {
    "reference": "Patient/44938004"
  },
  "dateTime": "2024-08-26T04:30:51.915104+00:00",
  "organization": [ {
    "reference": "Organization/example"
  } ],
  "sourceReference": {
    "reference": "QuestionnaireResponse/44938002"
  },
  "provision": {
    "type": "deny",
    "purpose": [ {
      "system": "http://purl.obolibrary.org/obo/duo.owl",
      "code": "RS_PD"
    } ]
  }
}
```

laptop : can take more than one provision in a consent resource.



consent\_yes.json > ...

```
1  {
2    "resourceType": "Consent",
3    "id": "consent-yes",
4    "status": "active",
5    "scope": {
6      "coding": [
7        {
8          "system": "http://terminology.hl7.org/CodeSystem/consentscope",
9          "code": "research",
10         "display": "Research"
11       }
12     ],
13   },
14   "category": [
15     {
16       "coding": [
17         {
18           "system": "http://terminology.hl7.org/CodeSystem/consentcategory",
19           "code": "dna",
20           "display": "DNA analysis"
21         }
22       ]
23     }
24   ],
25   "patient": {
26     "reference": "Patient/44938004"
27   },
28   "dateTime": "2024-08-26T04:30:51.915104+00:00",
29   "organization": [
30     {
31       "reference": "Organization/example"
32     }
33   ],
34   "sourceReference": {
35     "reference": "QuestionnaireResponse/44938002"
36   },
37   "provision": [
38     {
39       "type": "deny",
40       "purpose": [
41         {
42           "system": "http://purl.obolibrary.org/obo/duo.owl",
43           "code": "RS_PD"
44         }
45       ]
46     },
47     {
48       "type": "permit",
49       "purpose": [
50         {
51           "system": "http://purl.obolibrary.org/obo/duo.owl",
52           "code": "RS_POP"
53         }
54       ]
55     }
56   ]
57 }
```

If I had time, I would make these changes :

- Create an Organization Resource, to add it in our Consent Resource with the server ID.
- Better organize the items of the questionnaire, and adding the additional informations I asked for in the main\_function in the questionnaire (named : complete\_questionnaire.json)
- In my consent ressource, I can add only one provision by consent in the FHIR server. (but in my laptop, I can have other provisions added. try it with another server?)
- The duo code that is in some sub\_items doesn't appear. (like 4.1 : GRU, 4.2 : HMB..), i've got to modify how to make the questions, or the function.