#rarediseasesandcancershackathon

X-EHEALTH

3 - 4 OCTOBER 2022 · ONLINE EVENT

hackathon

RARE DISEASES AND CANCERS

SPMS EPE
Serviços Partilhados do Ministério da Saúde

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 951938

X-eHealth
Exchanging Electronic Health Records in a common framework

X-EHEALTH
hackathon
RARE DISEASES AND CANCERS

Blockchain based Notarization

Health Information Management – Austria

Dr. Christian Baumann (for MoH Austria)

#rarediseasesandcancershackathon

Agenda

- Blockchain based Notarization (general)
- Notarization of machine generated data
  - (in the context of e-Health)
  - Processes
  - Components of the prototype/PoC
  - Implementation
- PoC Assessment Criteria

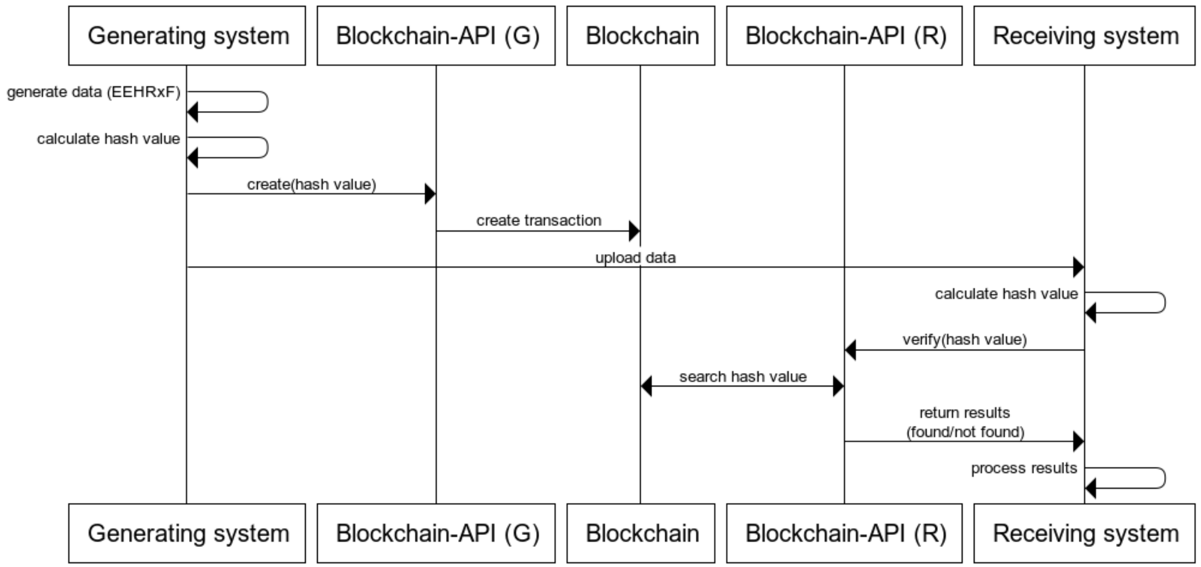
Blockchain based Notarization

- Notarization can be used to prove that an
  - electronic document
  - existed at a certain point in time
  - in a certain form and
  - has not been changed since then
- The security and trust that
  - notarized data cannot be manipulated
  - are guaranteed by blockchain technology
    - i.e. cryptographic functions hashing & digital signatures
- Only anonymous data is processed!
  - i.e. hash values of electronic documents
  - NO personal data, NO health data “on-chain”

Notarization of machine generated data 1/2

- Generating system
  - Generates data (e.g. in EEHRxF) *
  - Notarizes data
    - calculates hash value and stores it on blockchain
  - Sends data to receiving system
- Receiving system
  - Verifies integrity of data
    - calculates hash value, looks it up on blockchain
    - checks timestamp and public key of sender
  - Processes data

Notarization of machine generated data

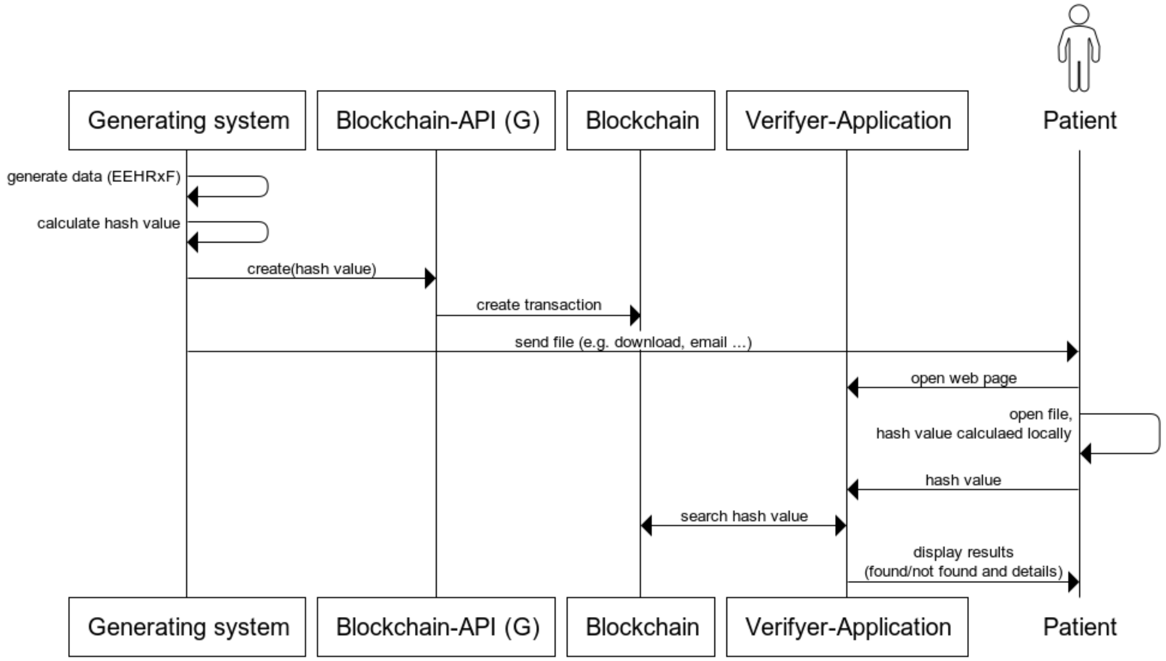


(*) ALL formats can be used: HL7, IHE, ICD-*, Snomed, Loinc …, PDF, Logfiles, DB dumps …

Notarization of machine generated data 2/2

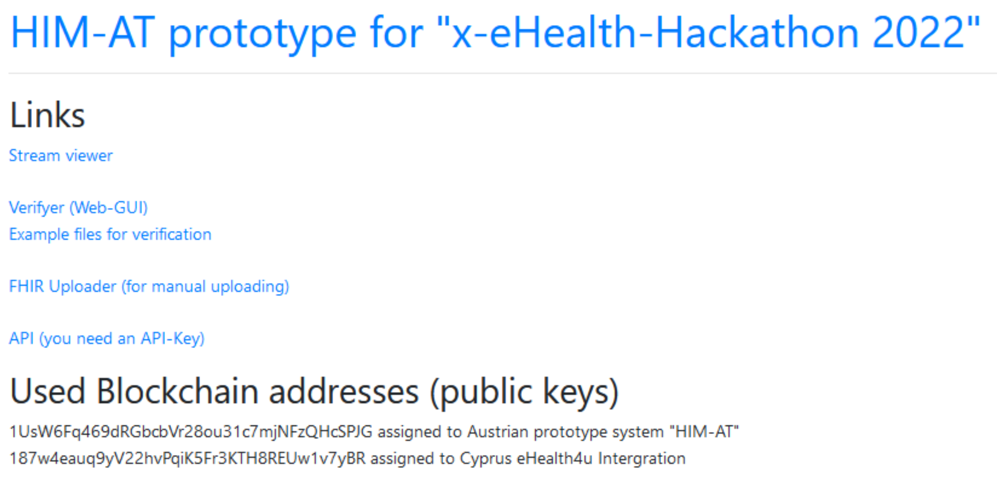
- “Receiving system” = patient
  - Documents e.g.
    - Lab reports
    - Sickness certificates
  - Verifies integrity of data
    - Using a Web-GUI

Notarization of machine generated data - manual verification



Components (provided/implemented)

- Access to a blockchain system (already existing)
  - Special „stream“ for the data to be processed in the context of this hackathon
- REST-API for notarization and verifcation
  - example scripts and API keys for interested parties
- Prototype scripts for
  - generating/sending system
  - receiving/processing system
- Web-GUIs for
  - viewing the blockchain stream
  - manual verification of files
- Prototype: https://test.baumann.at/xeh/



HIM-AT prototype for "x-eHealth-Hackathon 2022"

Links

Stream viewer

Verifyer (Web-GUI)
Example files for verification

FHIR Uploader (for manual uploading)

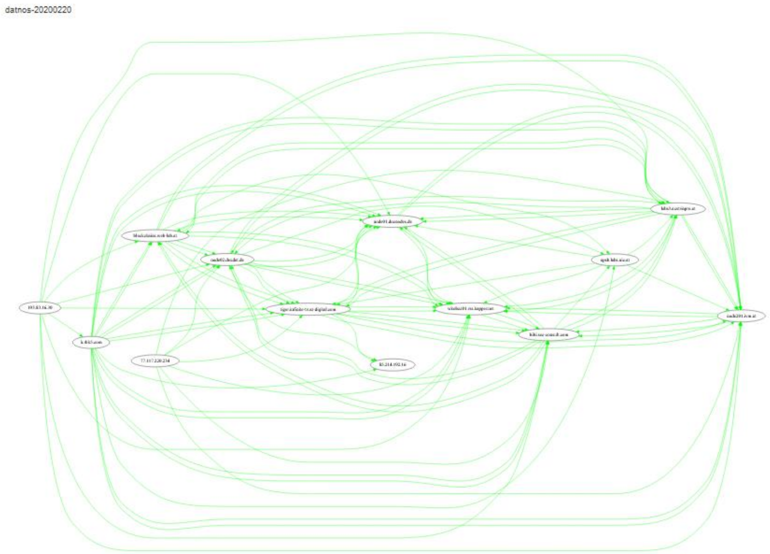
API (you need an API-Key)

Used Blockchain addresses (public keys)

1UsW6Fq469dRGbcbVr28ou31c7mjNFzQHcSPJG assigned to Austrian prototype system "HIM-AT"
187w4eauq9yV22hvPqiK5Fr3KTH8REUw1v7yBR assigned to Cyprus eHealth4u Intergration

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 951938

Blockchain System



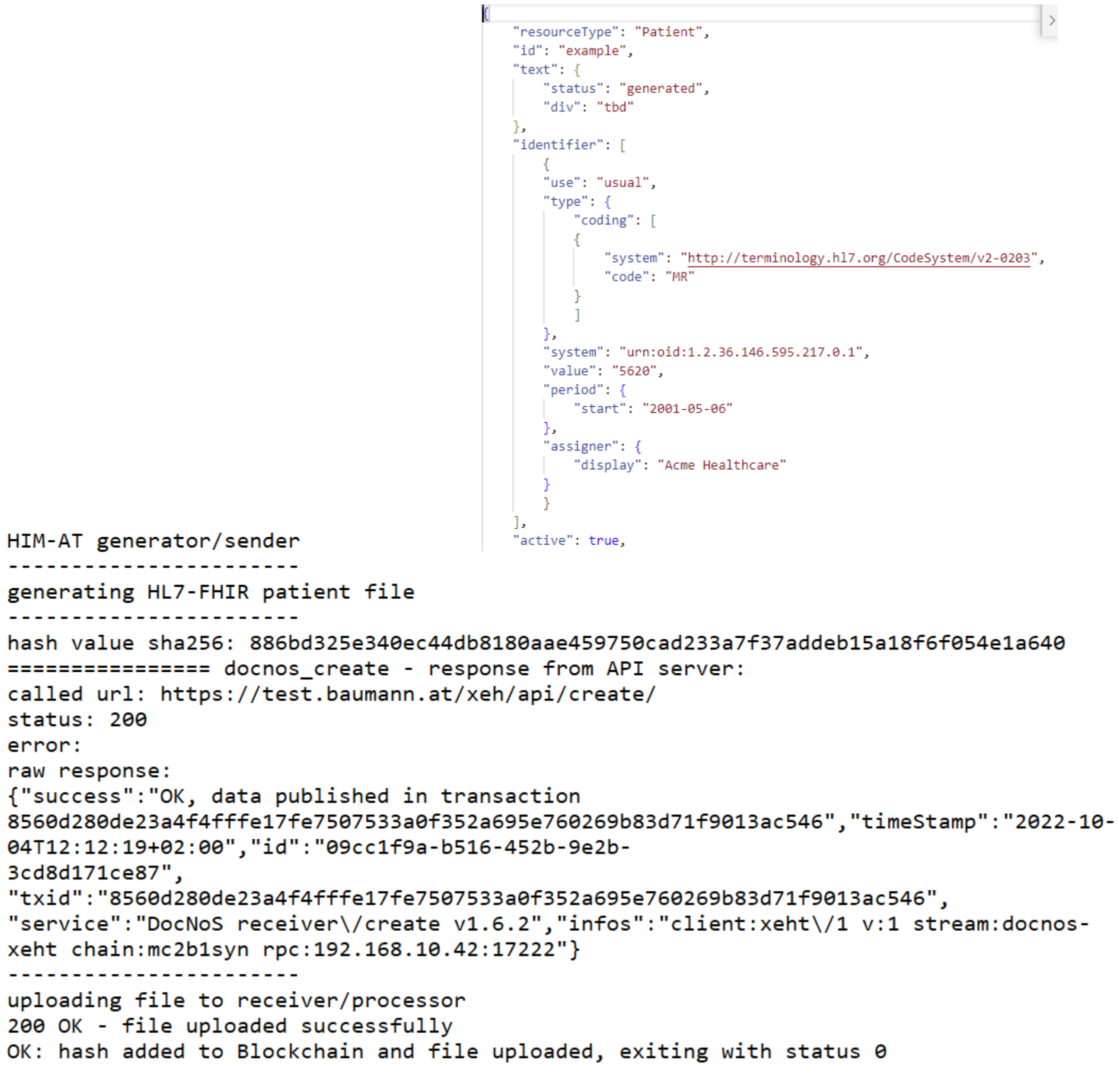
- Existing system („DatNoS“-chain)
  - different instances: Test-chains, production …
  - ca. 12 nodes in operation
  - Proof of Authority, only identifies parties, low energy!
- Based on „Multichain“ (multichain.com)
  - opensource
  - rapid deployment, data-streams, fine-grained permissions, highly customizable
  - own „stream“ for the data beeing processed in this hackathon

REST-API

- Connects applications to the blockchain
  - „create“ … adds a notarization record
  - „verify“ … searches stream for notarization records (hash values , document-ids …)
- Multi-Client
  - allows configuration for many clients (applications)
  - each with own API-key and other parameters
- Example scripts available
  - Python
  - (viz. Cyprus eHealth4u Integration - Discharge Summary)

Generating/sending system

- Generates data
  - i.e. HL7/FHIR patient records
  - saves data locally
- Notarizes the data by
  - calculating the hash value and calling the API
- Transmits the data to the receiving system
  - i.e. https-upload

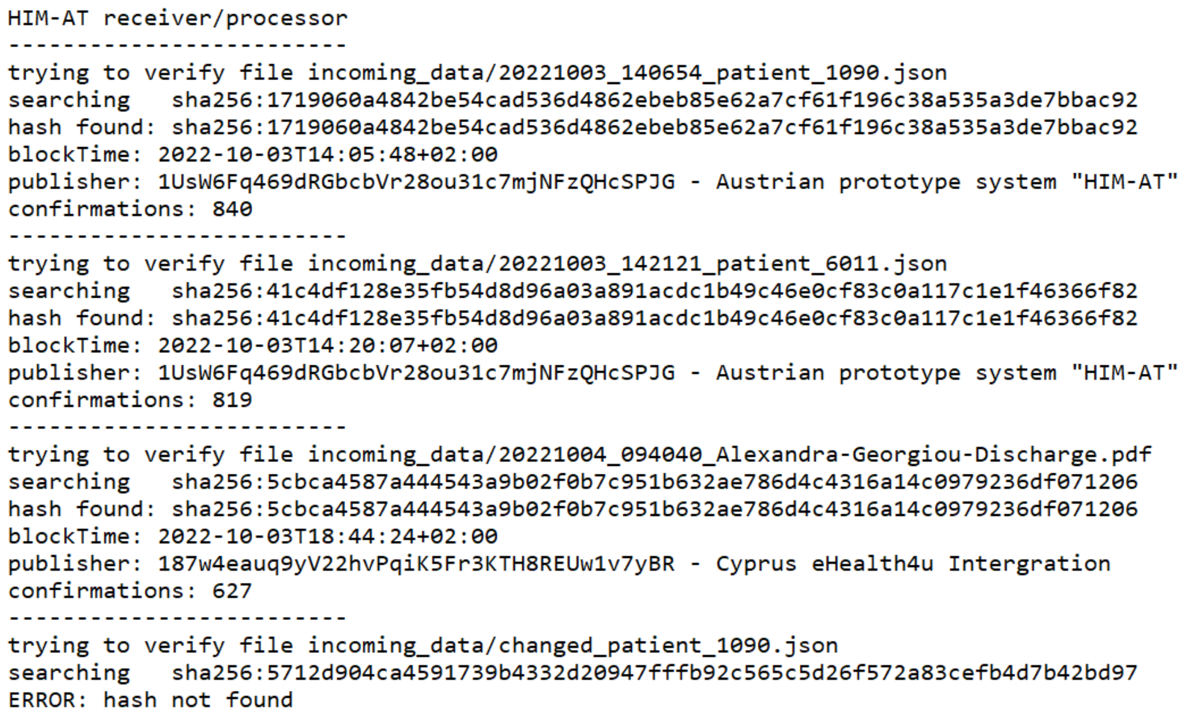


```
{
    "resourceType": "Patient",
    "id": "example",
    "text": {
        "status": "generated",
        "div": "tbd"
    },
    "identifier": [
        {
        "use": "usual",
        "type": {
            "coding": [
            {
                "system": "http://terminology.hl7.org/CodeSystem/v2-0203",
                "code": "MR"
            }
            ]
        },
        "system": "urn:oid:1.2.36.146.595.217.0.1",
        "value": "5620",
        "period": {
            "start": "2001-05-06"
        },
        "assigner": {
            "display": "Acme Healthcare"
        }
        }
    ],
    "active": true,
```

```
HIM-AT generator/sender
-----------------------
generating HL7-FHIR patient file
-----------------------
hash value sha256: 886bd325e340ec44db8180aae459750cad233a7f37addeb15a18f6f054e1a640
================ docnos_create - response from API server:
called url: https://test.baumann.at/xeh/api/create/
status: 200
error:
raw response:
{"success":"OK, data published in transaction
8560d280de23a4f4fffe17fe7507533a0f352a695e760269b83d71f9013ac546","timeStamp":"2022-10-
04T12:12:19+02:00","id":"09cc1f9a-b516-452b-9e2b-
3cd8d171ce87",
"txid":"8560d280de23a4f4fffe17fe7507533a0f352a695e760269b83d71f9013ac546",
"service":"DocNoS receiver\/create v1.6.2","infos":"client:xeht\/1 v:1 stream:docnos-
xeht chain:mc2b1syn rpc:192.168.10.42:17222"}
-----------------------
uploading file to receiver/processor
200 OK - file uploaded successfully
OK: hash added to Blockchain and file uploaded, exiting with status 0
```

Receiving/processing system

- Endpoint for https-upload
- Verifies data by
  - calculating the hash value of the received file
  - and calling the API
  - evaluates the result
- Further processing of the data
- …

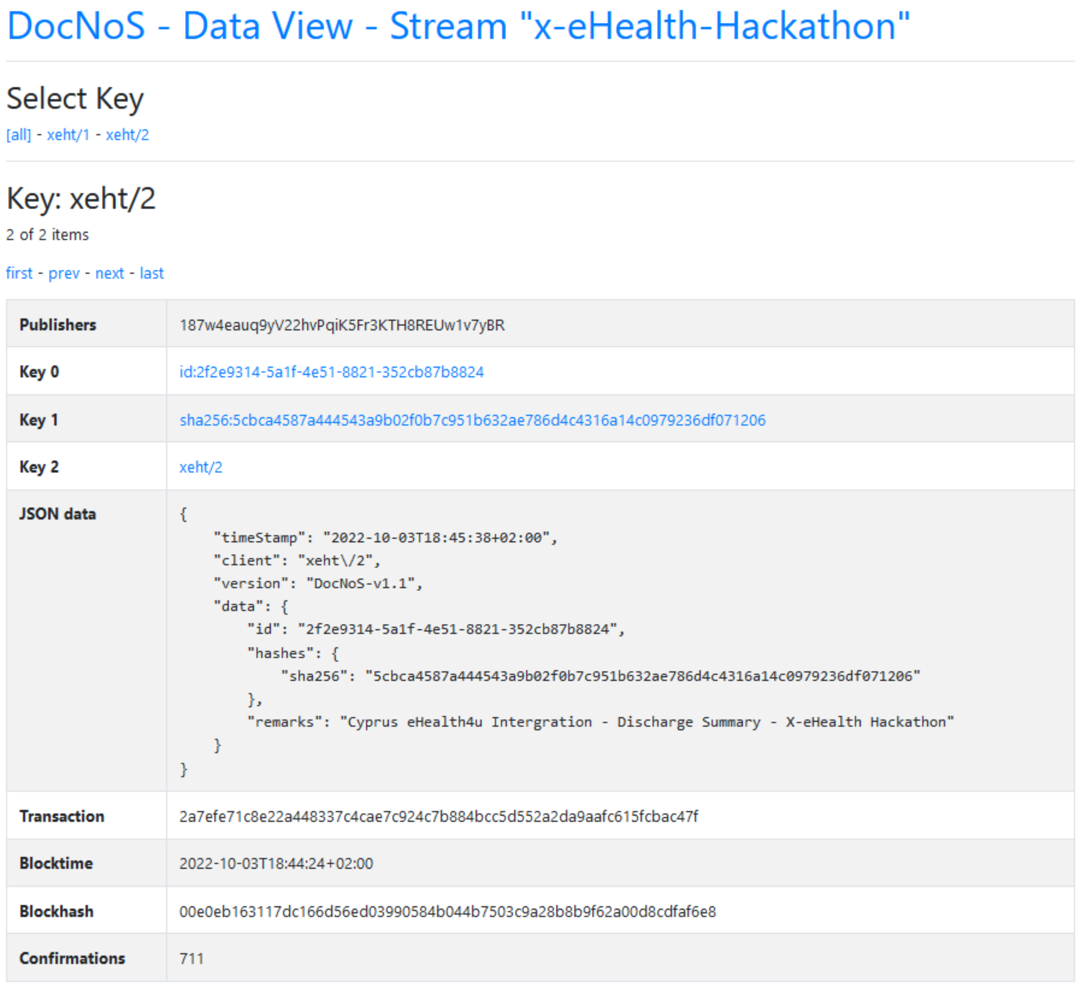


```
HIM-AT receiver/processor
-------------------------
trying to verify file incoming_data/20221003_140654_patient_1090.json
searching   sha256:1719060a4842be54cad536d4862ebeb85e62a7cf61f196c38a535a3de7bbac92
hash found: sha256:1719060a4842be54cad536d4862ebeb85e62a7cf61f196c38a535a3de7bbac92
blockTime: 2022-10-03T14:05:48+02:00
publisher: 1UsW6Fq469dRGbcbVr28ou31c7mjNFzQHcSPJG - Austrian prototype system "HIM-AT"
confirmations: 840
-------------------------
trying to verify file incoming_data/20221003_142121_patient_6011.json
searching   sha256:41c4df128e35fb54d8d96a03a891acdc1b49c46e0cf83c0a117c1e1f46366f82
hash found: sha256:41c4df128e35fb54d8d96a03a891acdc1b49c46e0cf83c0a117c1e1f46366f82
blockTime: 2022-10-03T14:20:07+02:00
publisher: 1UsW6Fq469dRGbcbVr28ou31c7mjNFzQHcSPJG - Austrian prototype system "HIM-AT"
confirmations: 819
-------------------------
trying to verify file incoming_data/20221004_094040_Alexandra-Georgiou-Discharge.pdf
searching   sha256:5cbca4587a444543a9b02f0b7c951b632ae786d4c4316a14c0979236df071206
hash found: sha256:5cbca4587a444543a9b02f0b7c951b632ae786d4c4316a14c0979236df071206
blockTime: 2022-10-03T18:44:24+02:00
publisher: 187w4eauq9yV22hvPqiK5Fr3KTH8REUw1v7yBR - Cyprus eHealth4u Intergration
confirmations: 627
-------------------------
trying to verify file incoming_data/changed_patient_1090.json
searching   sha256:5712d904ca4591739b4332d20947fffb92c565c5d26f572a83cefb4d7b42bd97
ERROR: hash not found
```

Web-GUI 1

Viewing the blockchain stream

- filter by keys
  - client
  - document-id
  - ...



DocNoS - Data View - Stream "x-eHealth-Hackathon"

Select Key

[all] - xeht/1 - xeht/2

Key: xeht/2

2 of 2 items

first - prev - next - last

Publishers	187w4eauq9yV22hvPqiK5Fr3KTH8REUw1v7yBR
Key 0	id:2f2e9314-5a1f-4e51-8821-352cb87b8824
Key 1	sha256:5cbca4587a444543a9b02f0b7c951b632ae786d4c4316a14c0979236df071206
Key 2	xeht/2

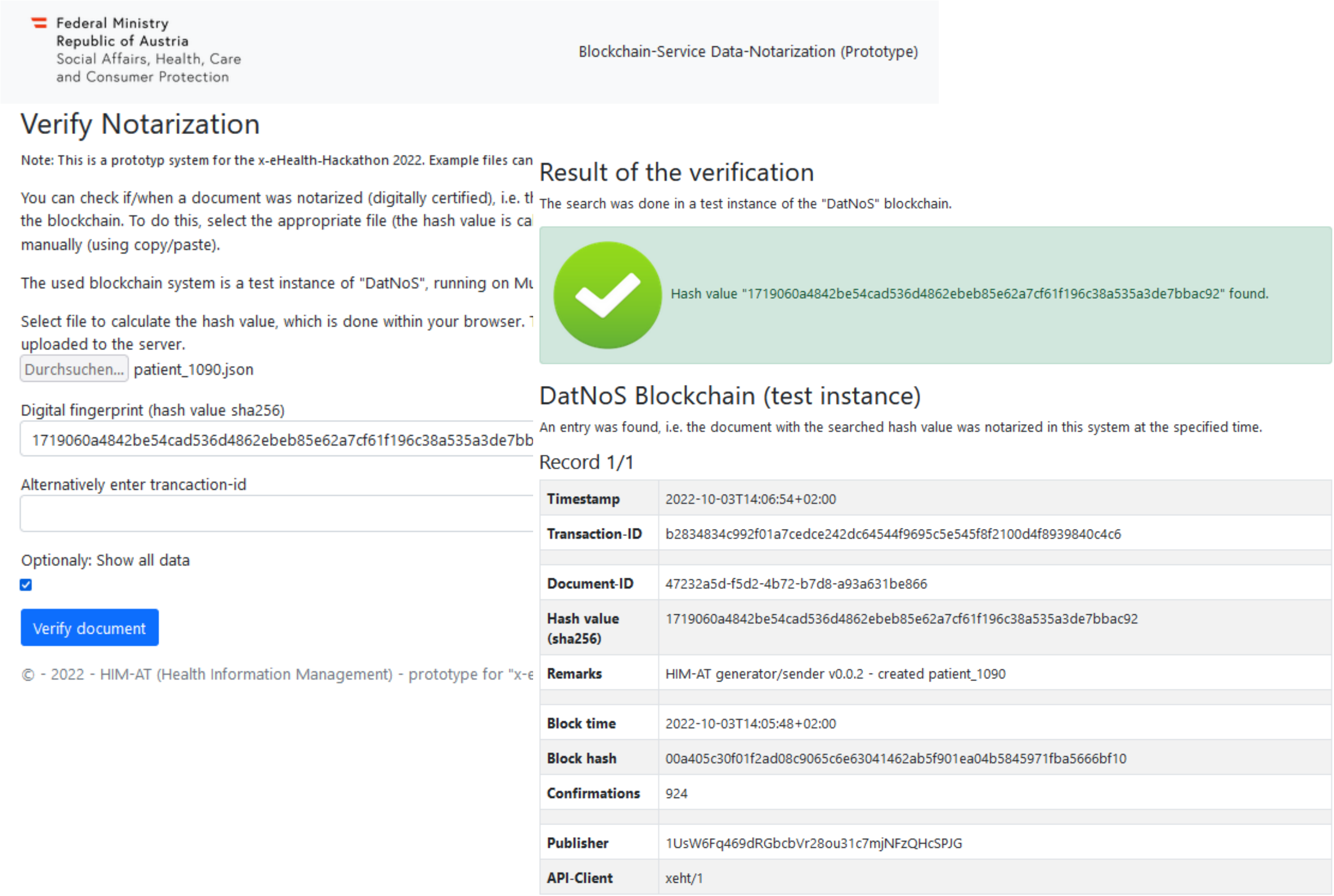
JSON data	{
	"timeStamp": "2022-10-03T18:45:38+02:00",
	"client": "xeht\/2",
	"version": "DocNoS-v1.1",
	"data": {
	"id": "2f2e9314-5a1f-4e51-8821-352cb87b8824",
	"hashes": {
	"sha256": "5cbca4587a444543a9b02f0b7c951b632ae786d4c4316a14c0979236df071206"
	},
	"remarks": "Cyprus eHealth4u Intergration - Discharge Summary - X-eHealth Hackathon"
	}
	}

Transaction	2a7efe71c8e22a448337c4cae7c924c7b884bcc5d552a2da9aafc615fcbac47f
Blocktime	2022-10-03T18:44:24+02:00
Blockhash	00e0eb163117dc166d56ed03990584b044b7503c9a28b8b9f62a00d8cdfaf6e8
Confirmations	711

Web-GUI 2

Manual verification of files

- open file in browser
- hash value is calculated locally (in the browser), file NOT uploaded to a server
- result & details are displayed



Federal Ministry
Republic of Austria
Social Affairs, Health, Care
and Consumer Protection

Blockchain-Service Data-Notarization (Prototype)

Verify Notarization

Note: This is a prototyp system for the x-eHealth-Hackathon 2022. Example files can

You can check if/when a document was notarized (digitally certified), i.e. tl
the blockchain. To do this, select the appropriate file (the hash value is cal
manually (using copy/paste).

The used blockchain system is a test instance of "DatNoS", running on Mu

Select file to calculate the hash value, which is done within your browser. T
uploaded to the server.

Durchsuchen... patient_1090.json

Digital fingerprint (hash value sha256)

1719060a4842be54cad536d4862ebeb85e62a7cf61f196c38a535a3de7bb

Alternatively enter trancaction-id

Optionaly: Show all data

☑

Verify document

© - 2022 - HIM-AT (Health Information Management) - prototype for "x-e

Result of the verification

The search was done in a test instance of the "DatNoS" blockchain.

✓ Hash value "1719060a4842be54cad536d4862ebeb85e62a7cf61f196c38a535a3de7bbac92" found.

DatNoS Blockchain (test instance)

An entry was found, i.e. the document with the searched hash value was notarized in this system at the specified time.

Record 1/1

Timestamp	2022-10-03T14:06:54+02:00
Transaction-ID	b2834834c992f01a7cedce242dc64544f9695c5e545f8f2100d4f8939840c4c6
Document-ID	47232a5d-f5d2-4b72-b7d8-a93a631be866
Hash value (sha256)	1719060a4842be54cad536d4862ebeb85e62a7cf61f196c38a535a3de7bbac92
Remarks	HIM-AT generator/sender v0.0.2 - created patient_1090
Block time	2022-10-03T14:05:48+02:00
Block hash	00a405c30f01f2ad08c9065c6e63041462ab5f901ea04b5845971fba5666bf10
Confirmations	924
Publisher	1UsW6Fq469dRGbcbVr28ou31c7mjNFzQHcSPJG
API-Client	xeht/1

Live demo …

- https://test.baumann.at/xeh/
- Python client
  - https://test.baumann.at/xeh/python-client/python-client-X-eHealth-Hackathon.zip
  - If you have any question … contact me

PoC Assessment Criteria 1/2

- Innovation
  - first prototype for blockchain based notarization for e-Health data (afaik)
- Impact
  - machine generated data can be notarized and verified automatically
  - trust between different systems can be increased (intentional or unintentional data changes can be detected)
  - patients can verify their documents manually
- Applicability
  - any form of data exchange in e-Health sector

PoC Assessment Criteria 2/2

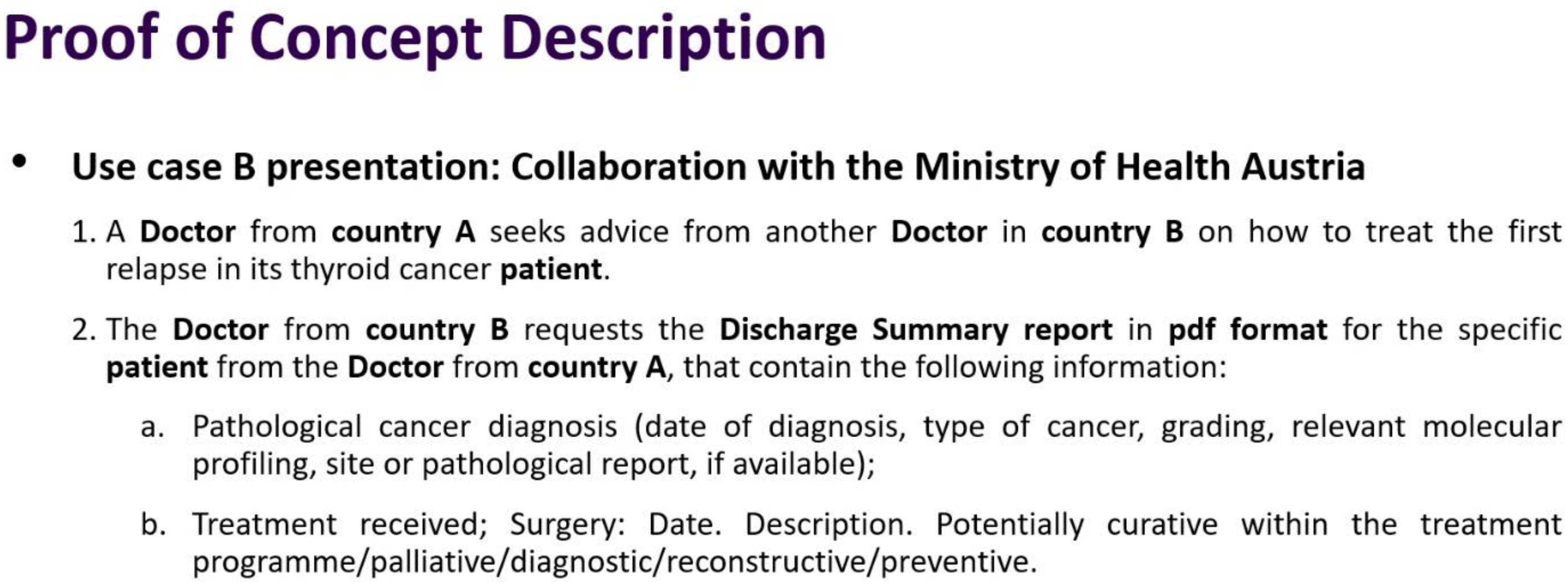
- Feasibility
  - proven by this prototype, more details to be analyzed
- Technical background
  - standardized data and file formats used: HL7, IHE, ICD-*, Snomed, Loinc …, PDF
  - standardized interfaces (REST, https, json …)
  - open source blockchain system Multichain (any other blockchain system can also be used)
- Contribution to EEHRxF Development
  - extend/expand the prototype system for more interested parties?

Questions?

Appendix

- Integration of
  - Cyprus eHealth4U PDF Discharge Summary with
  - Notarization in HIM-AT prototype system

Cyprus eHealth4U
PDF Notarization of Discharge Summary



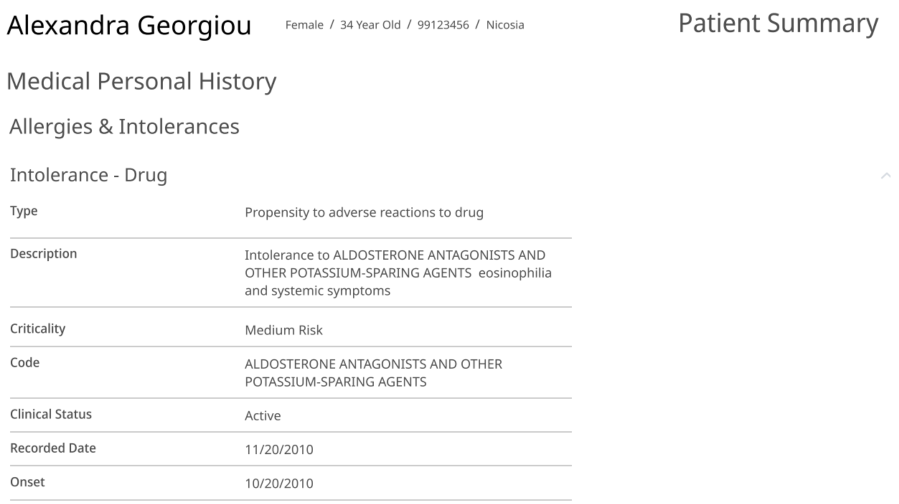
Proof of Concept Description

- **Use case B presentation: Collaboration with the Ministry of Health Austria**

1. A **Doctor** from **country A** seeks advice from another **Doctor** in **country B** on how to treat the first relapse in its thyroid cancer **patient**.
2. The **Doctor** from **country B** requests the **Discharge Summary report** in **pdf format** for the specific **patient** from the **Doctor** from **country A**, that contain the following information:
   a. Pathological cancer diagnosis (date of diagnosis, type of cancer, grading, relevant molecular profiling, site or pathological report, if available);
   b. Treatment received; Surgery: Date. Description. Potentially curative within the treatment programme/palliative/diagnostic/reconstructive/preventive.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 951938

Cyprus – Discharge Summary Report & Blockchain Notarization



Alexandra Georgiou Female / 34 Year Old / 99123456 / Nicosia

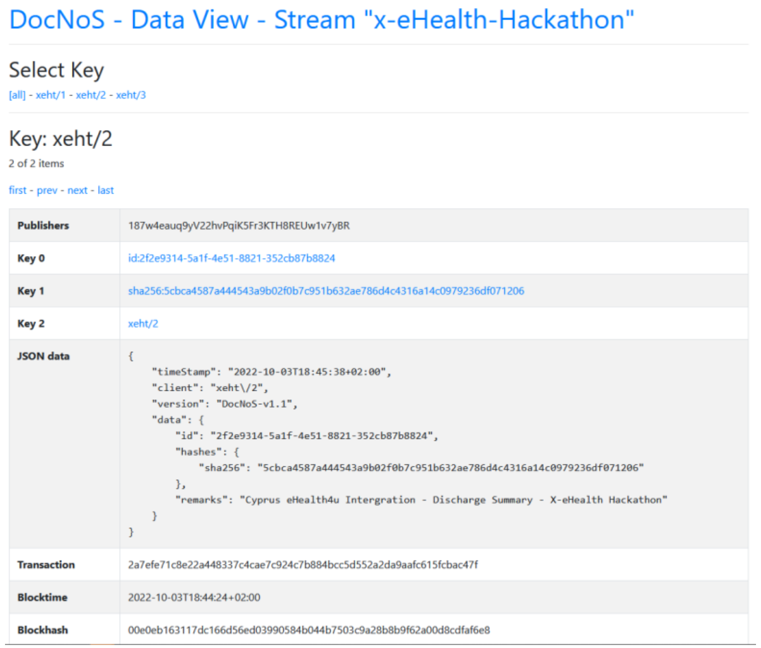
Patient Summary

Medical Personal History

Allergies & Intolerances

Intolerance - Drug

Type	Propensity to adverse reactions to drug
Description	Intolerance to ALDOSTERONE ANTAGONISTS AND OTHER POTASSIUM-SPARING AGENTS eosinophilia and systemic symptoms
Criticality	Medium Risk
Code	ALDOSTERONE ANTAGONISTS AND OTHER POTASSIUM-SPARING AGENTS
Clinical Status	Active
Recorded Date	11/20/2010
Onset	10/20/2010



DocNoS - Data View - Stream "x-eHealth-Hackathon"

Select Key

[all] - xeht/1 - xeht/2 - xeht/3

Key: xeht/2

2 of 2 items

first - prev - next - last

Publishers	187w4eauq9yV22hvPqiK5Fr3KTH8REUw1v7yBR
Key 0	id:2f2e9314-5a1f-4e51-8821-352cb87b8824
Key 1	sha256:5cbca4587a444543a9b02f0b7c951b632ae786d4c4316a14c0979236df071206
Key 2	xeht/2

JSON data	{
	"timeStamp": "2022-10-03T18:45:38+02:00",
	"client": "xeht\/2",
	"version": "DocNoS-v1.1",
	"data": {
	"id": "2f2e9314-5a1f-4e51-8821-352cb87b8824",
	"hashes": {
	"sha256": "5cbca4587a444543a9b02f0b7c951b632ae786d4c4316a14c0979236df071206"
	},
	"remarks": "Cyprus eHealth4u Intergration - Discharge Summary - X-eHealth Hackathon"
	}
	}

Transaction	2a7efe71c8e22a448337c4cae7c924c7b884bcc5d552a2da9aafc615fcbac47f
Blocktime	2022-10-03T18:44:24+02:00
Blockhash	00e0eb163117dc166d56ed03990584b044b7503c9a28b8b9f62a00d8cdfaf6e8

SPMS EPE
Serviços Partilhados do Ministério da Saúde

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 951938

X-eHealth
Exchanging Electronic Health Records in a common framework