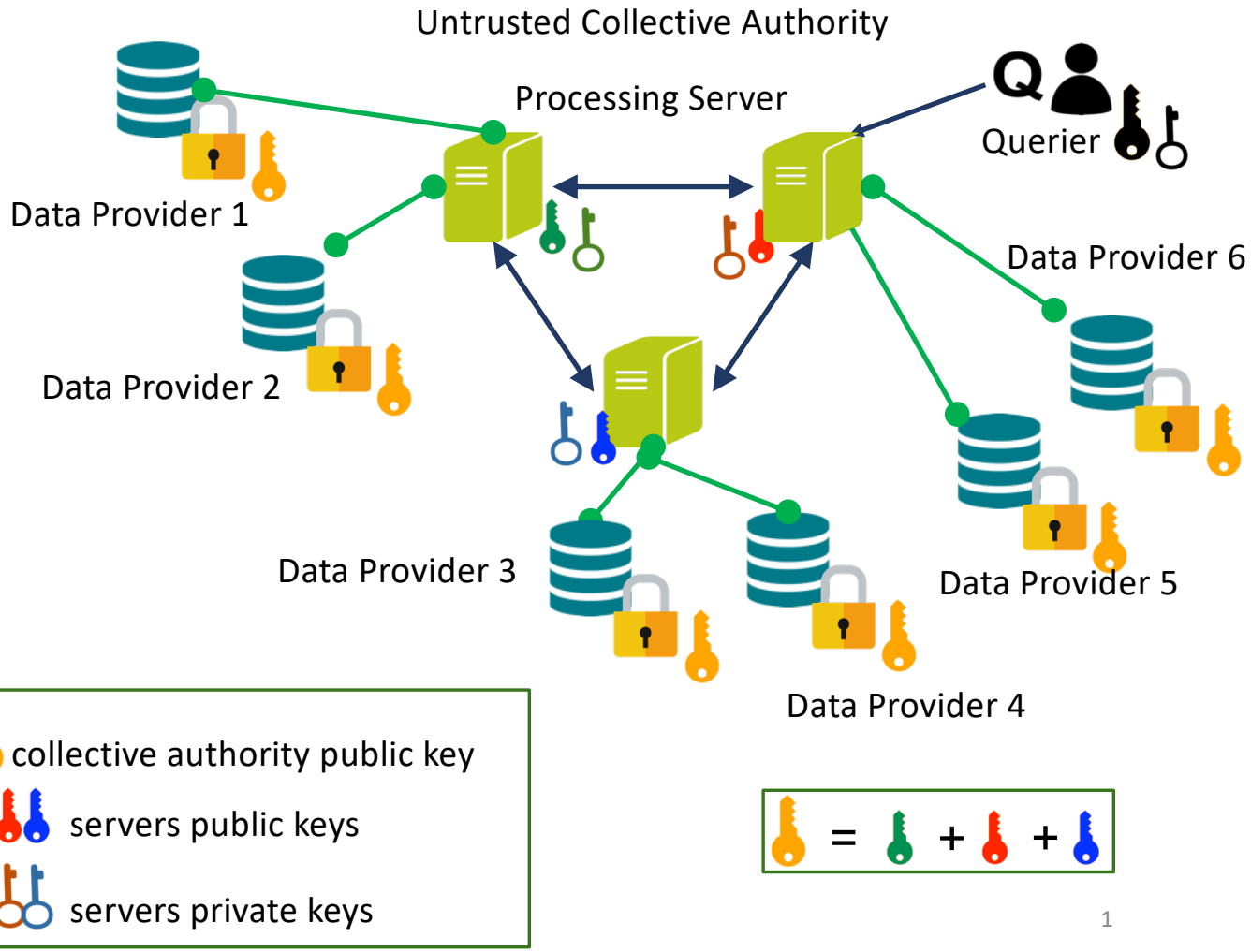
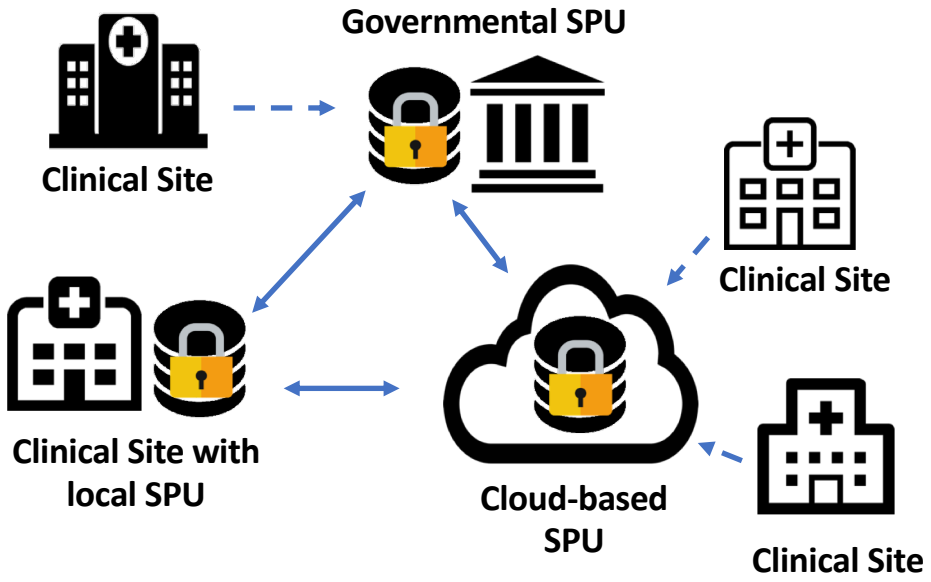
UnLynx: framework for privacy-conscious data sharing

[Froelicher et al PET'17]

- Trust is shared across a group of servers forming a collective authority
- They collaborate together to generate a collective encryption key
- The collective encryption key is used to encrypt the data and can be compromised only if all servers are compromised



MedCo: Privacy-conscious medical data sharing



Main features:

- **Secure outsourcing** enabled by **collective encryption** of the data
- **End-to-end data protection** through **homomorphic encryption**
- Compliance wrt to **regulations** (e.g., GDPR)
- **Increased flexibility and lower costs** wrt standard approaches of data sharing

Raisaro JL, Troncoso-Pastoriza JR, Misbach M, Gomes de Sá ES, André J, Pradervand S, Missiaglia E, Michielin O, Ford BA, Hubaux JP. **MedCo: Enabling Secure and Privacy-Preserving Exploration of Distributed Clinical and Genomic Data**. Accepted for publication in IEEE/ACM Transactions in Computational Biology and Bioinformatics

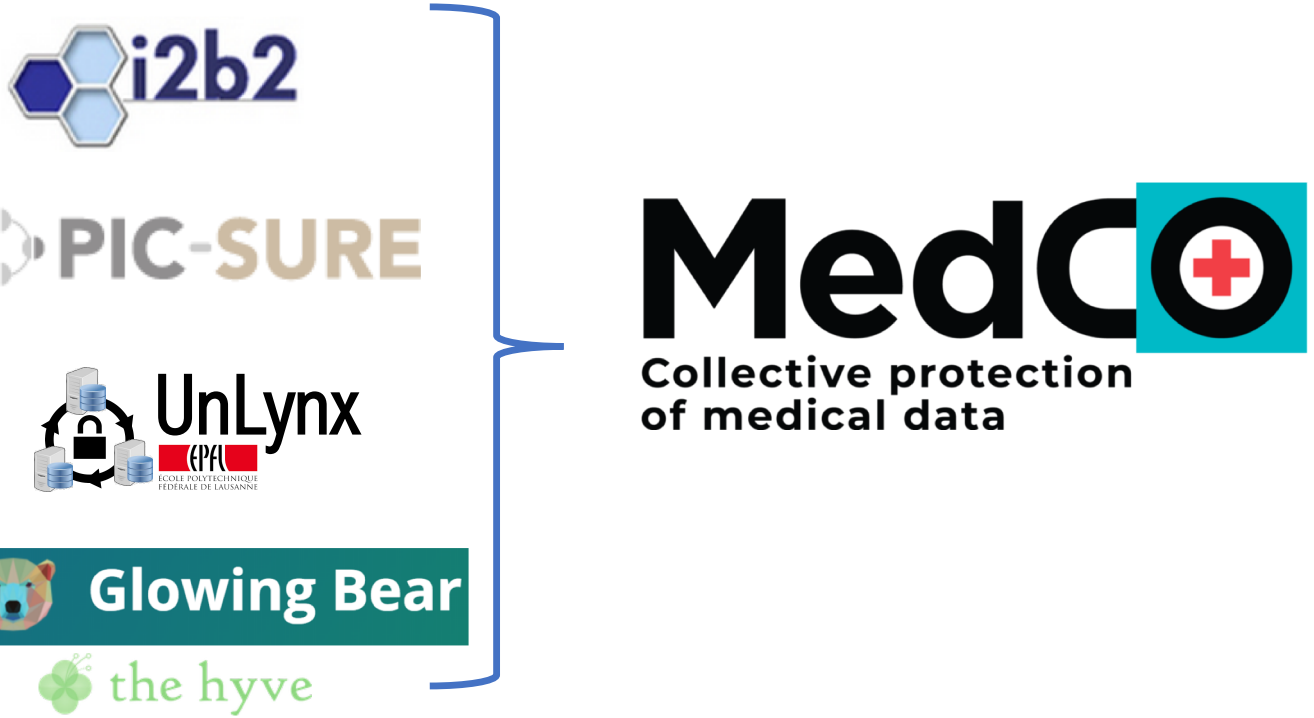
MedCo: Combining the best of Information Security and Medical Informatics

Data model

Interoperability layer
Meta API

Privacy-preserving computing framework

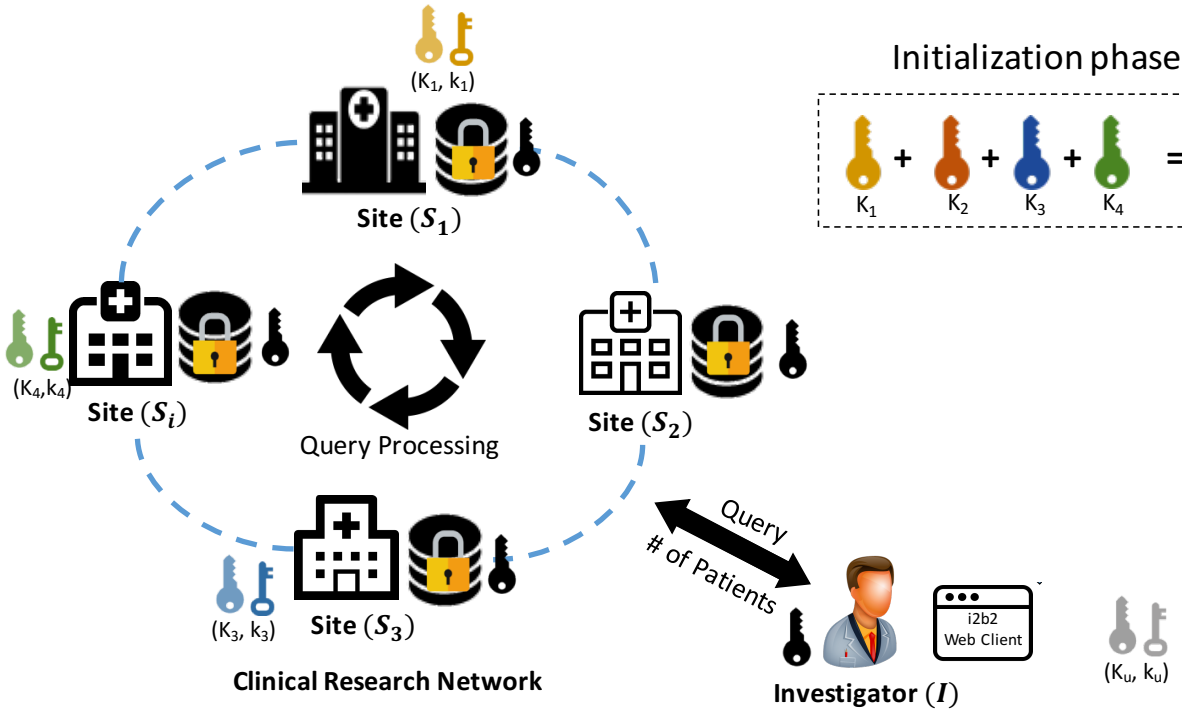
Modern GUI



DISCLAIMER

MedCo is a generic concept and it is not fundamentally tied to these technologies, but can be adapted and integrated to other ones

MedCo secure query protocol



Initialization phase

A, B) ETL & Encryption Phase

1) (user) Query Generation
2) (distributed) Query Tagging
3) (local) Query Processing
4) (local) Result Aggregation
5) (local) Result Obfuscation
6) (distributed) Results Shuffling
7) (distributed) Results Re-Encryption
8) (user) Result Decryption

ETL: extract, transform, load
(Database operations)

 Sites' public keys

 Sites' secret keys

 Collective public key

MedCo: ETL & encryption phase

- sensitive
- not sensitive

Concept Dimension – local i2b2

Concept_path	Concept_CD
path_1	code_1
path_2	code_2
path_3	code_3
path_4	code_4

Homomorphic Encryption

Concept Dimension

Concept_path	Concept_CD
griabfiyqeg	code_B
fgeiwbgohg	code_C
path_3	code_A
path_4	code_D

DDT

Concept Dimension - MedCo

Concept_path	Concept_CD
fagvfuafygf	code_B
tgybyggyyg	code_C
path_3	code_A
path_4	code_D

Observation Fact - MedCo

Patient_Num	Concept_CD
0000001	code_B
0000001	code_C
0000001	code_A
0000002	code_B
0000002	code_D
0000003	code_C
0000003	code_A

Patient Dimension – local i2b2

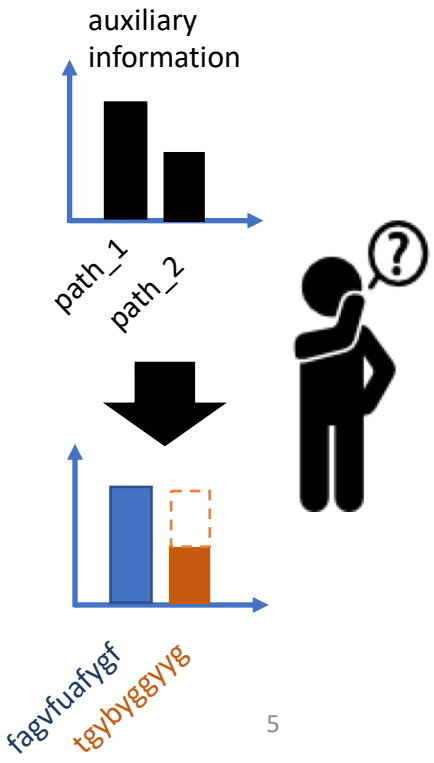
Patient_Num	isReal
0000001	1
0000002	1
0000003	0

Homomorphic Encryption

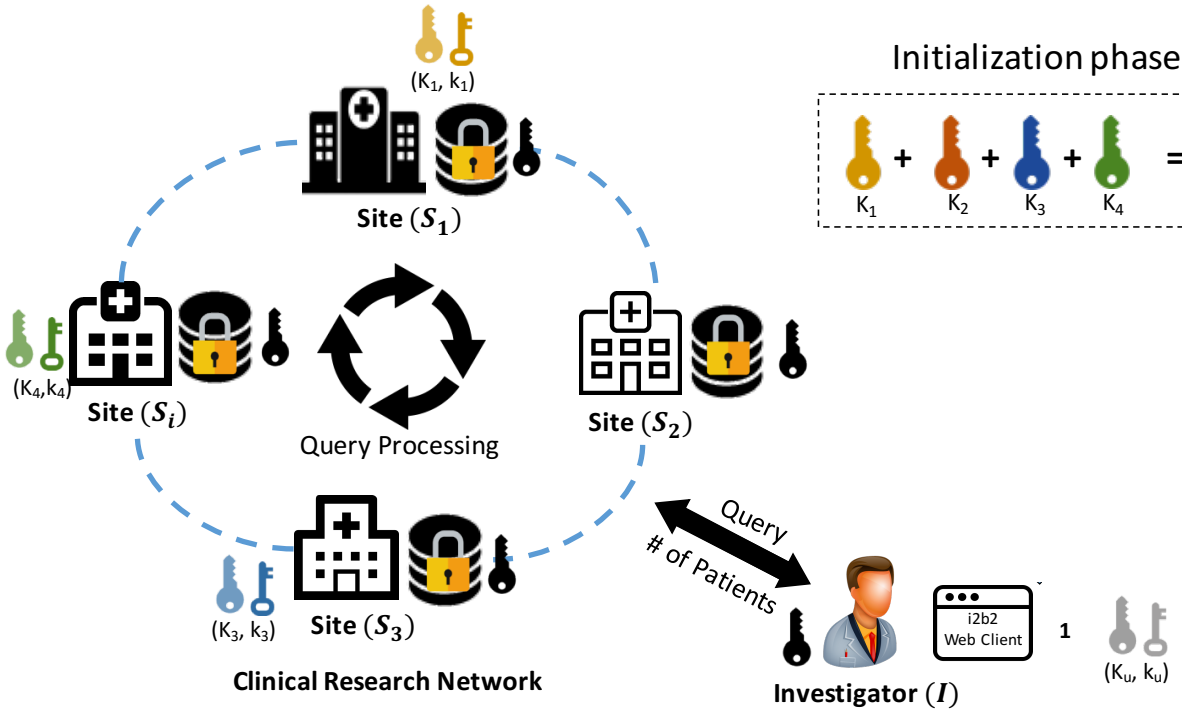
Patient Dimension - MedCo

Patient_Num	isReal
0000001	rteiqugfhb
0000002	vbwiygw=
0000003	fhbfg72=g

Frequency Attack!!



MedCo secure query protocol



Initialization phase

1) (user) Query Generation:

```
COUNT(DISTINCT(patient_Num))
FROM Observation Fact
WHERE Concept_path = path_2
AND Concept_path = path_3
...
```

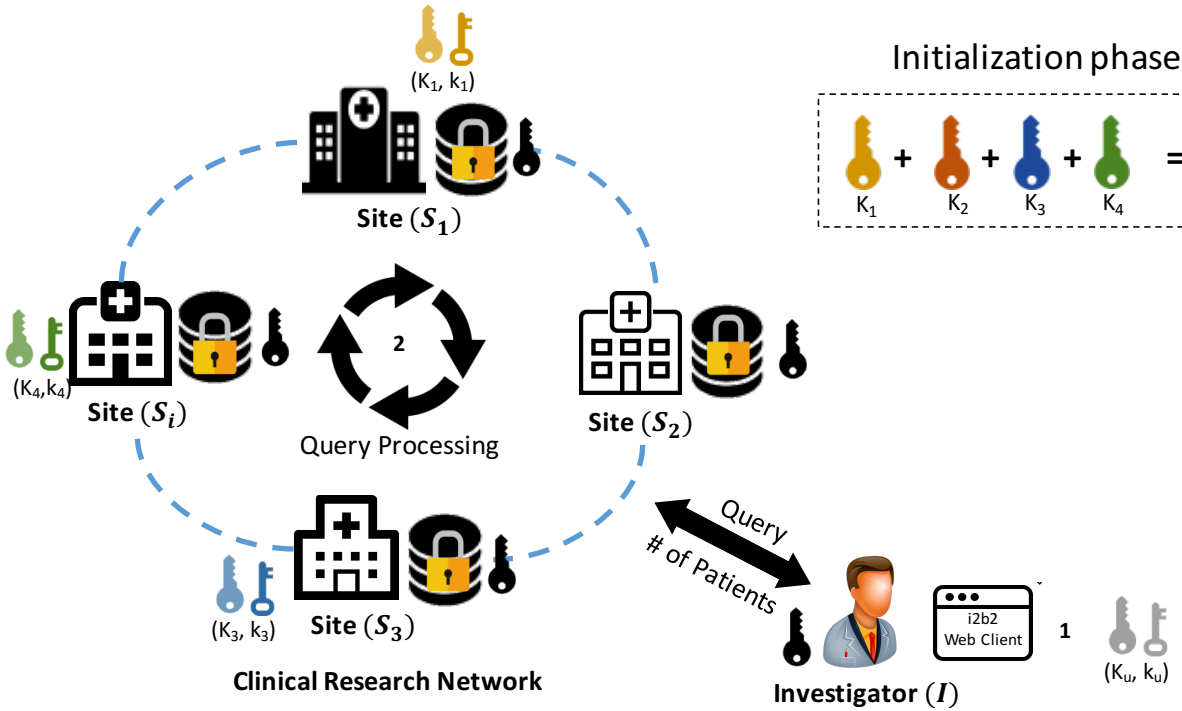

```
COUNT(DISTINCT(patient_Num))
FROM Observation Fact
WHERE Concept_path = hf78e2ib78ffg
AND Concept_path = path_3
...
```

 Sites' public keys

 Sites' secret keys

 Collective public key

MedCo secure query protocol



Initialization phase

2) (distributed) Query Tagging:

COUNT(DISTINCT(patient_Num))
FROM Observation Fact
WHERE Concept_path = hf78e2ib78ffg
AND Concept_path = path_3
…

DDT

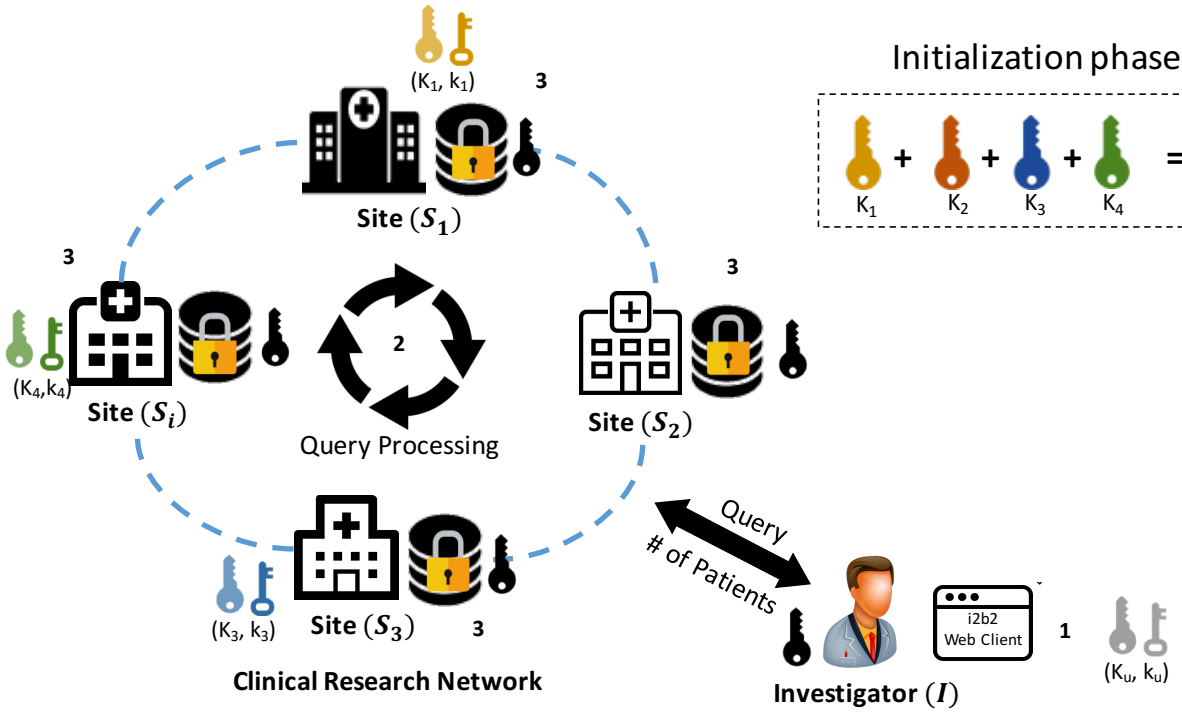
SELECT(DISTINCT(patient_Num))
FROM Observation Fact
WHERE Concept_path = tgybyggyyg
AND Concept_path = path_3
…

 Sites' public keys

 Sites' secret keys

 Collective public key

MedCo secure query protocol



Initialization phase

3) (local) Query Processing:

```
SELECT(DISTINCT(patient_Num))
FROM Observation Fact
WHERE Concept_path = tgybyggyyg
AND Concept_path = path_3
...
```

Observation Fact - MedCo

Patient_Num	Concept_CD
0000001	code_B
0000001	code_C
0000001	code_A
0000002	code_B
0000002	code_D
0000003	code_C
0000003	code_A

Concept Dimension - MedCo

Concept_path	Concept_CD
fagvfuafygf	code_B
tgybyggyyg	code_C
path_3	code_A
path_4	code_D

Patient Dimension - MedCo

Patient_Num	isReal
0000001	rteiqugfhb
0000002	vbwiygw=
0000003	fhbfg72=g

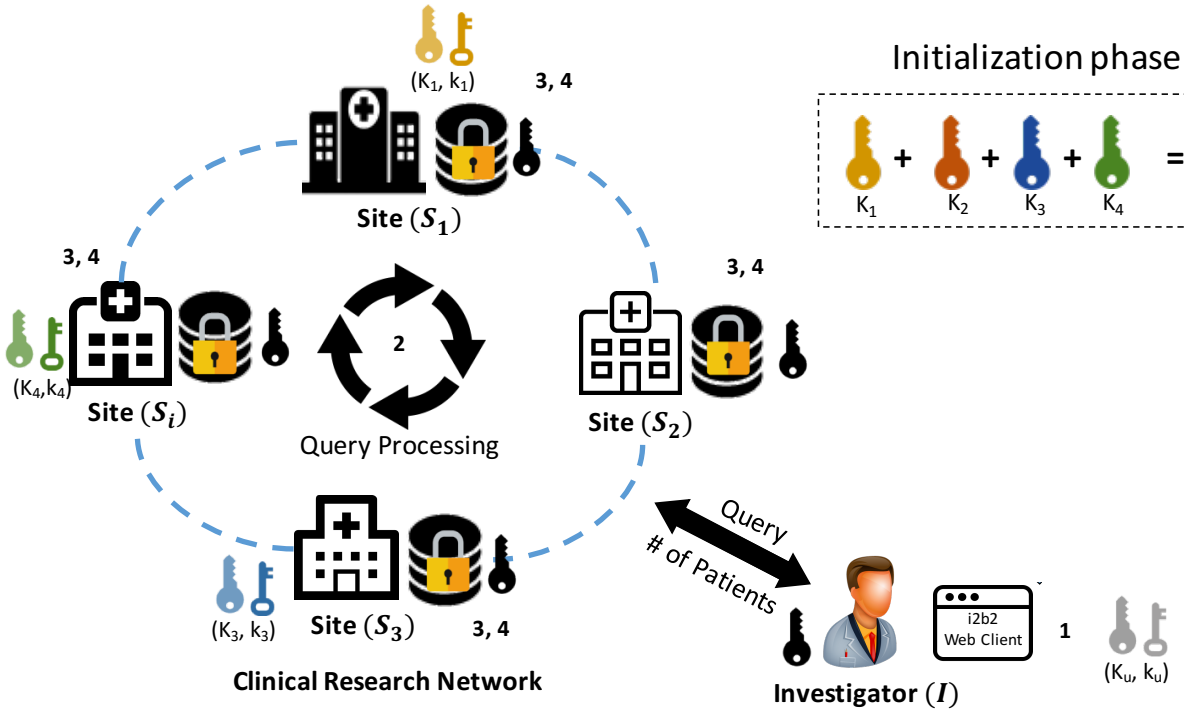
0000001	rteiqugfhb
0000003	fhbfg72=g

 Sites' public keys

 Sites' secret keys

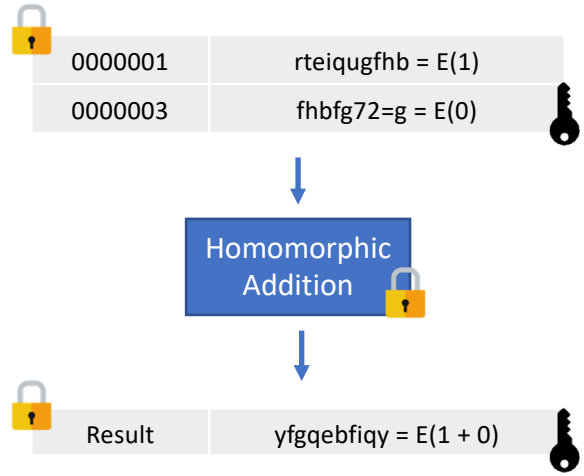
 Collective public key

MedCo secure query protocol



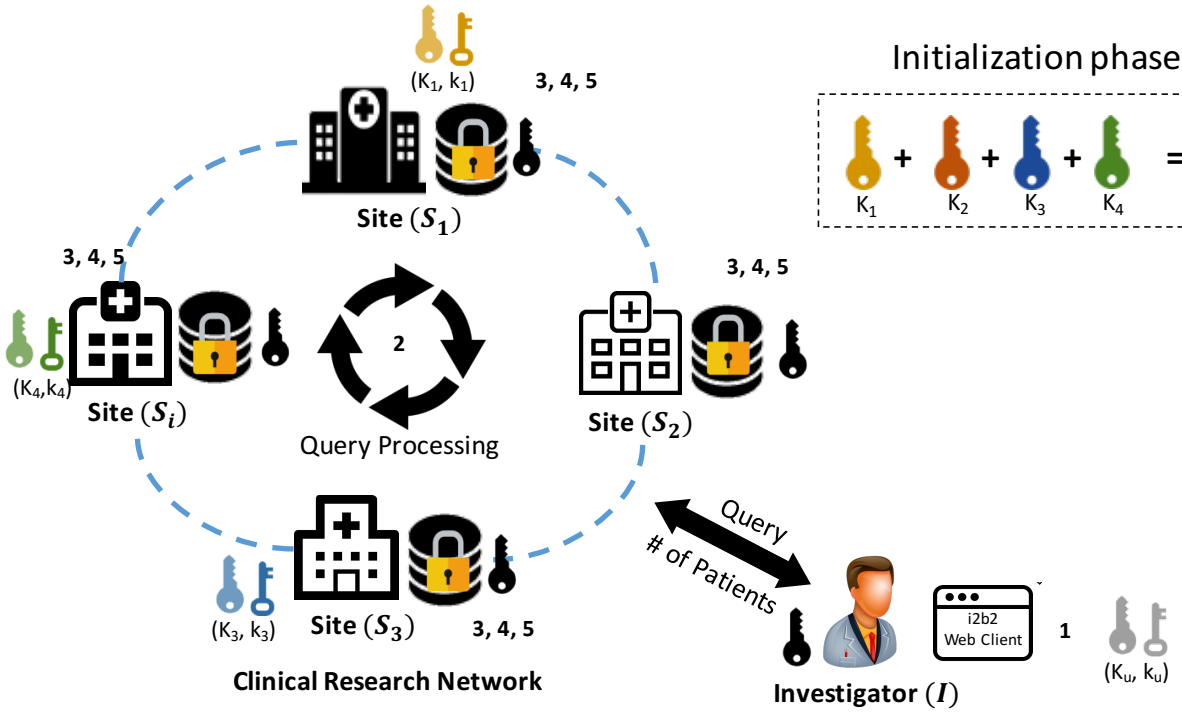
Initialization phase

4) (local) Result Aggregation:



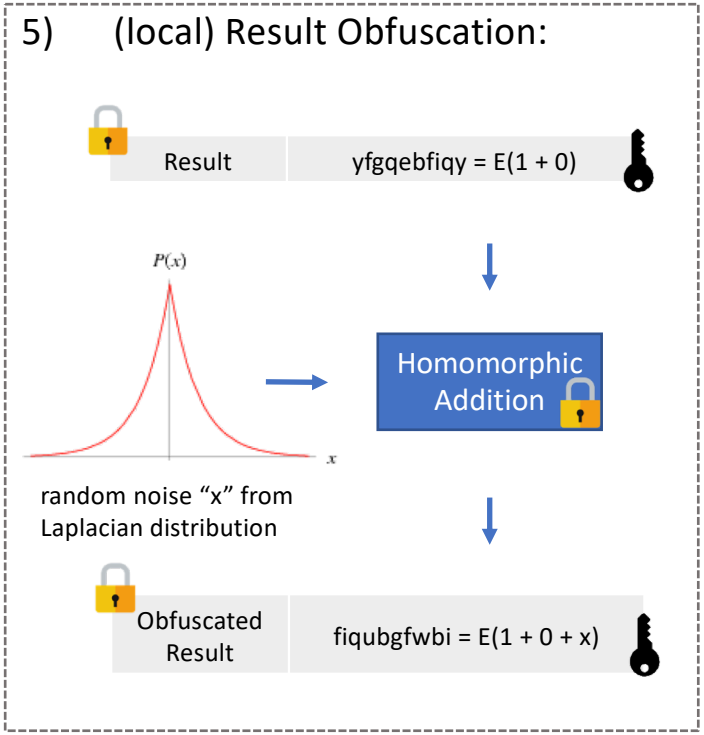
Sites' public keys Sites' secret keys Collective public key

MedCo secure query protocol



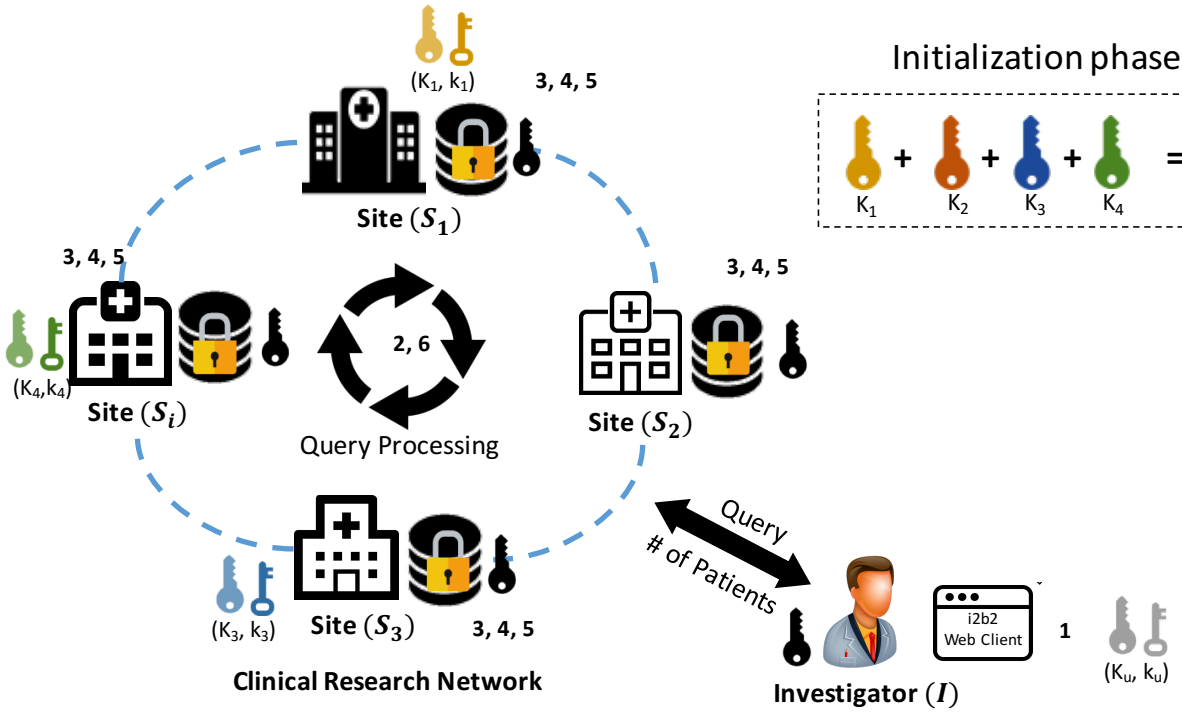
Initialization phase

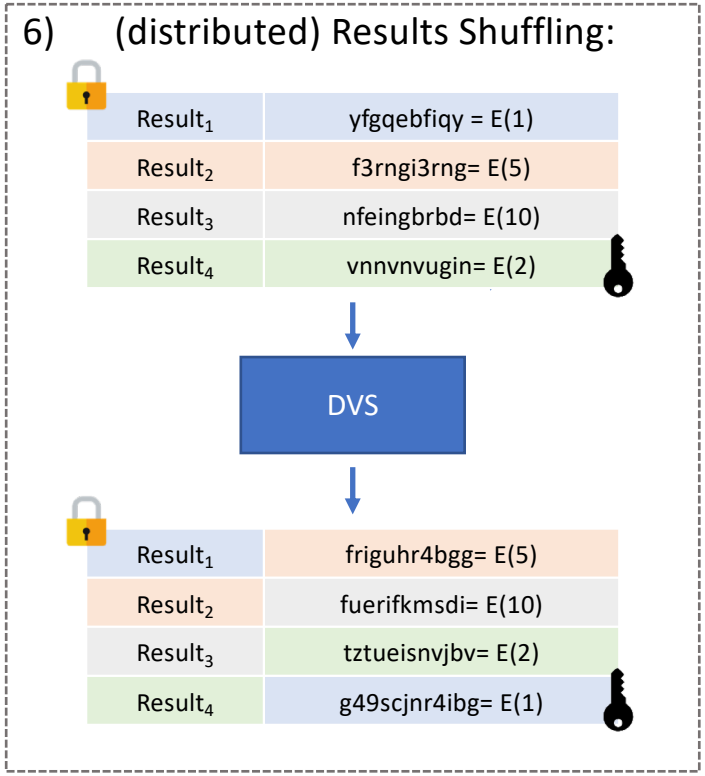
5) (local) Result Obfuscation:



Sites' public keys Sites' secret keys Collective public key

MedCo secure query protocol



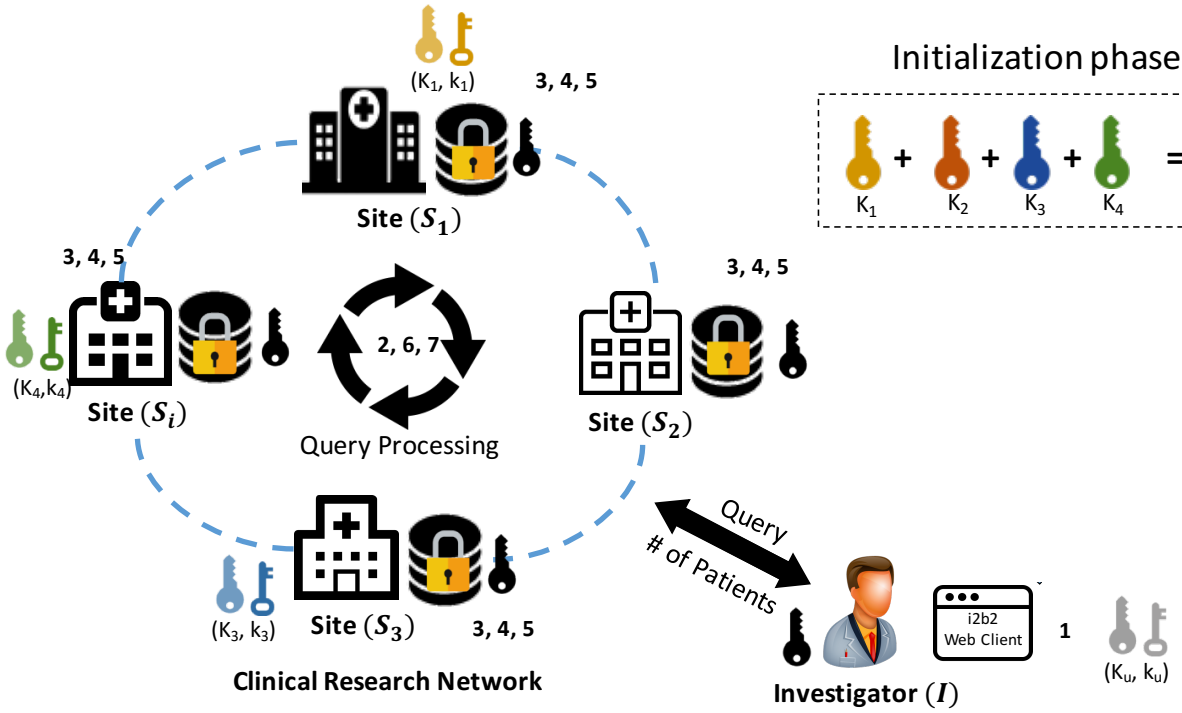


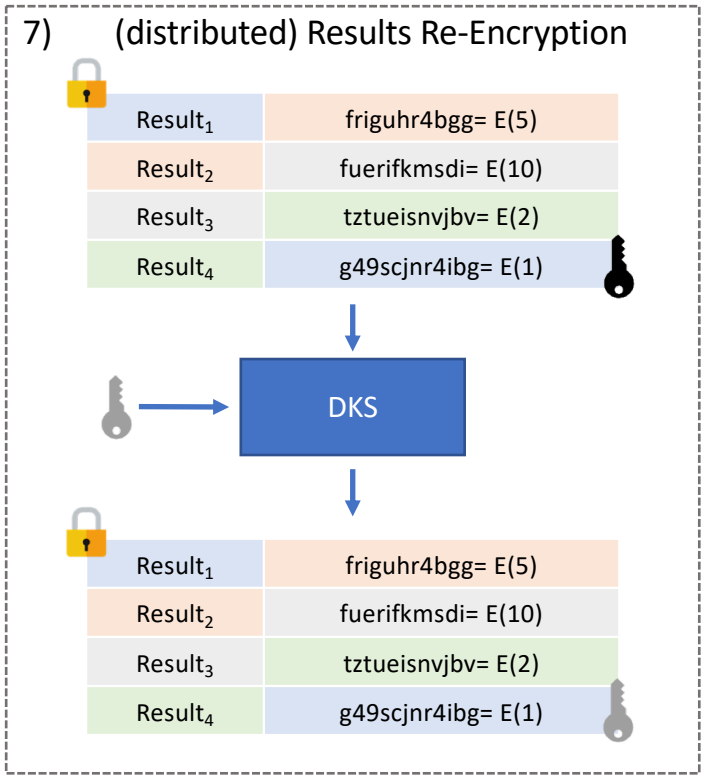
 Sites' public keys

 Sites' secret keys

 Collective public key

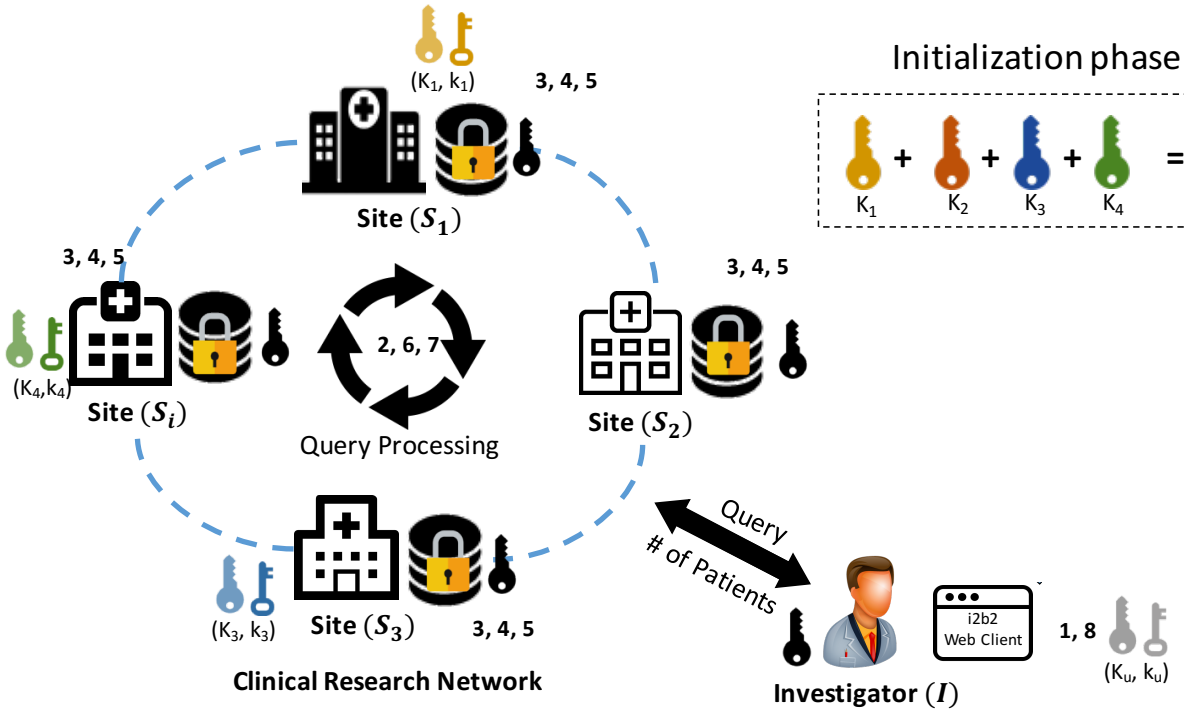
MedCo secure query protocol

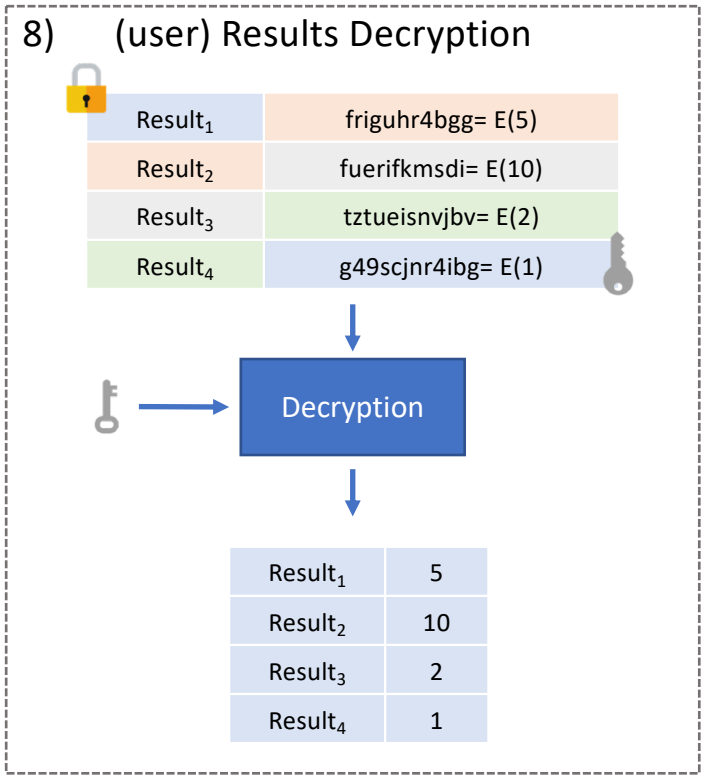




Sites' public keys Sites' secret keys Collective public key

MedCo secure query protocol



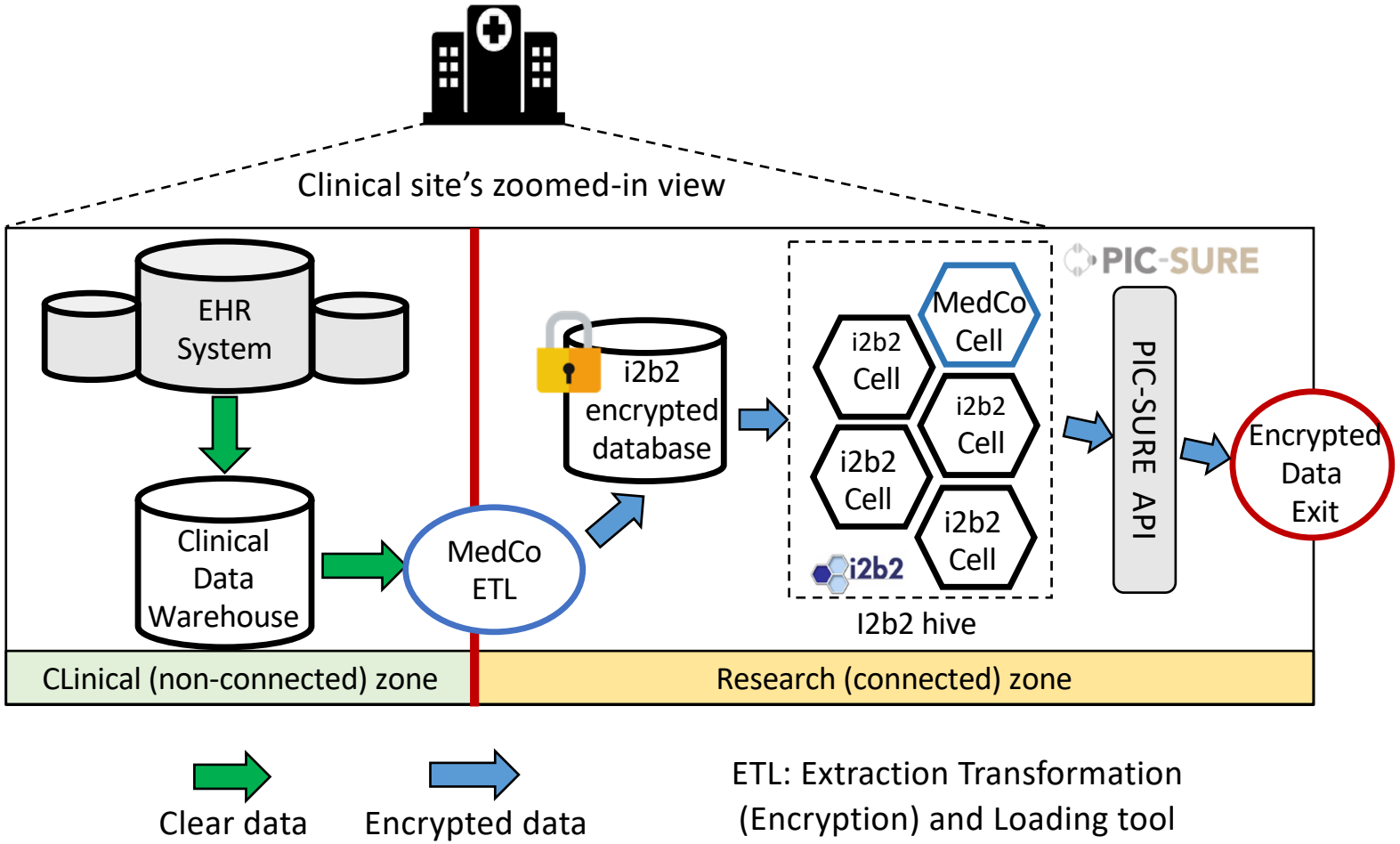


 Sites' public keys

 Sites' secret keys

 Collective public key

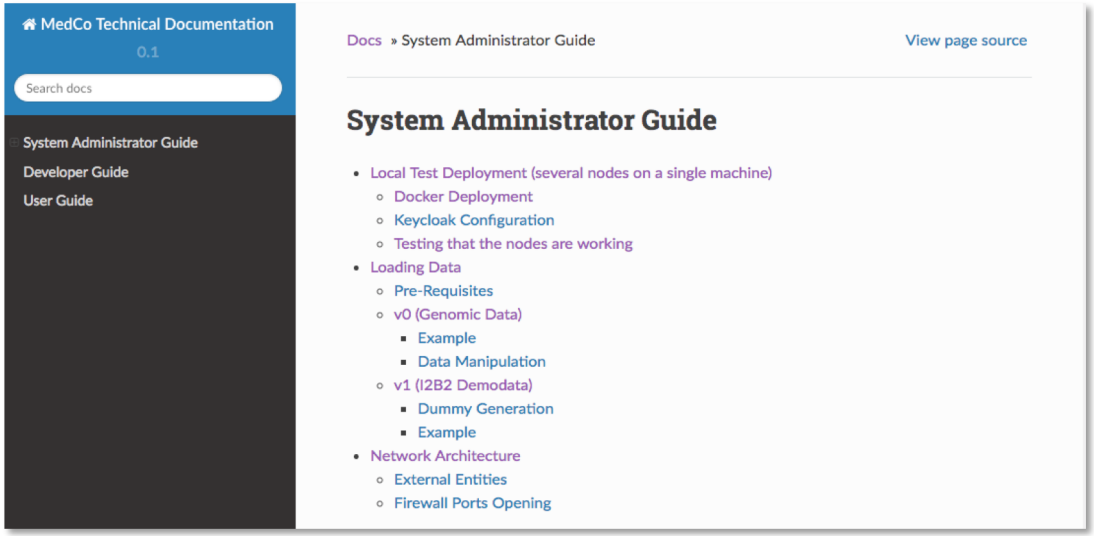
MedCo deployment at a clinical site



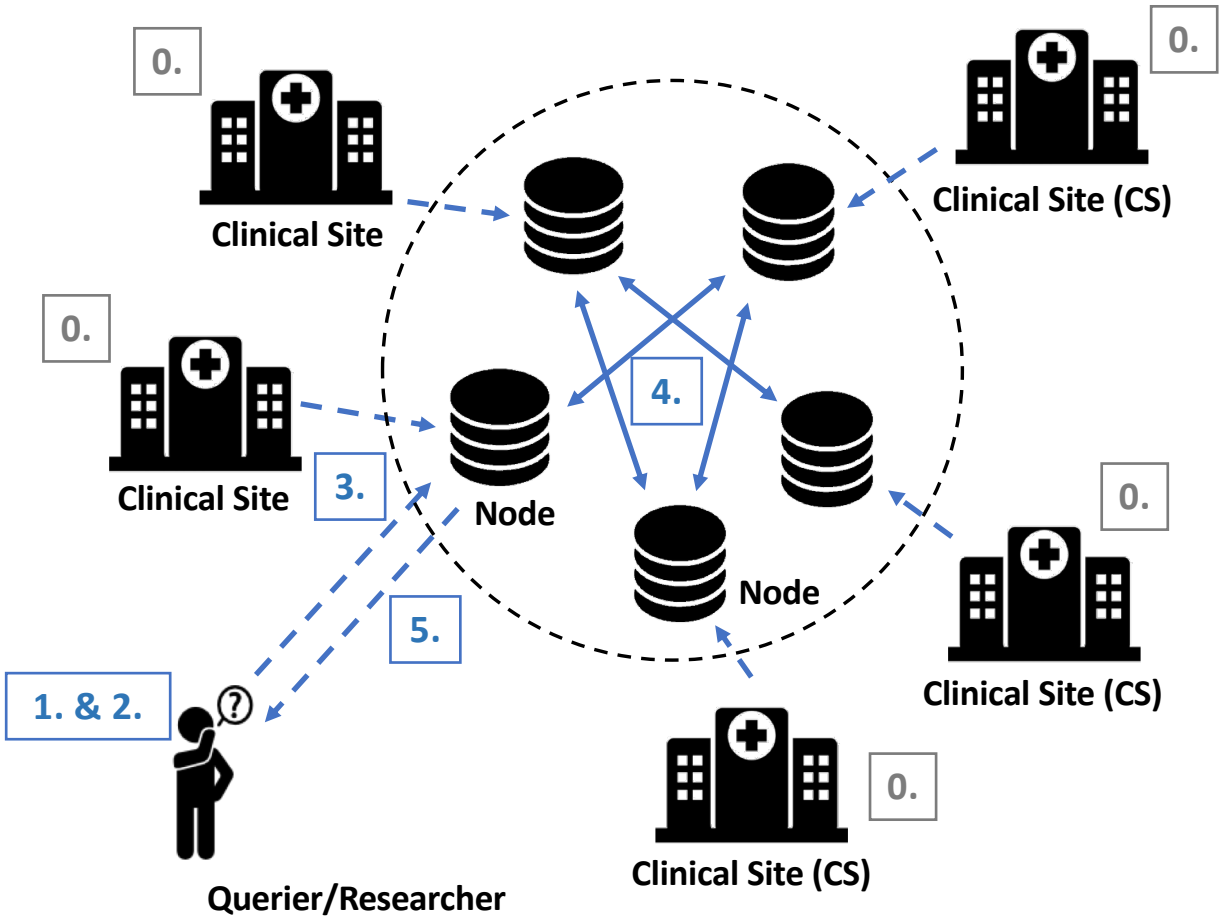
MedCo Web site and documentation

https://medco.epfl.ch/

https://lca1.github.io/medco-documentation/index.html



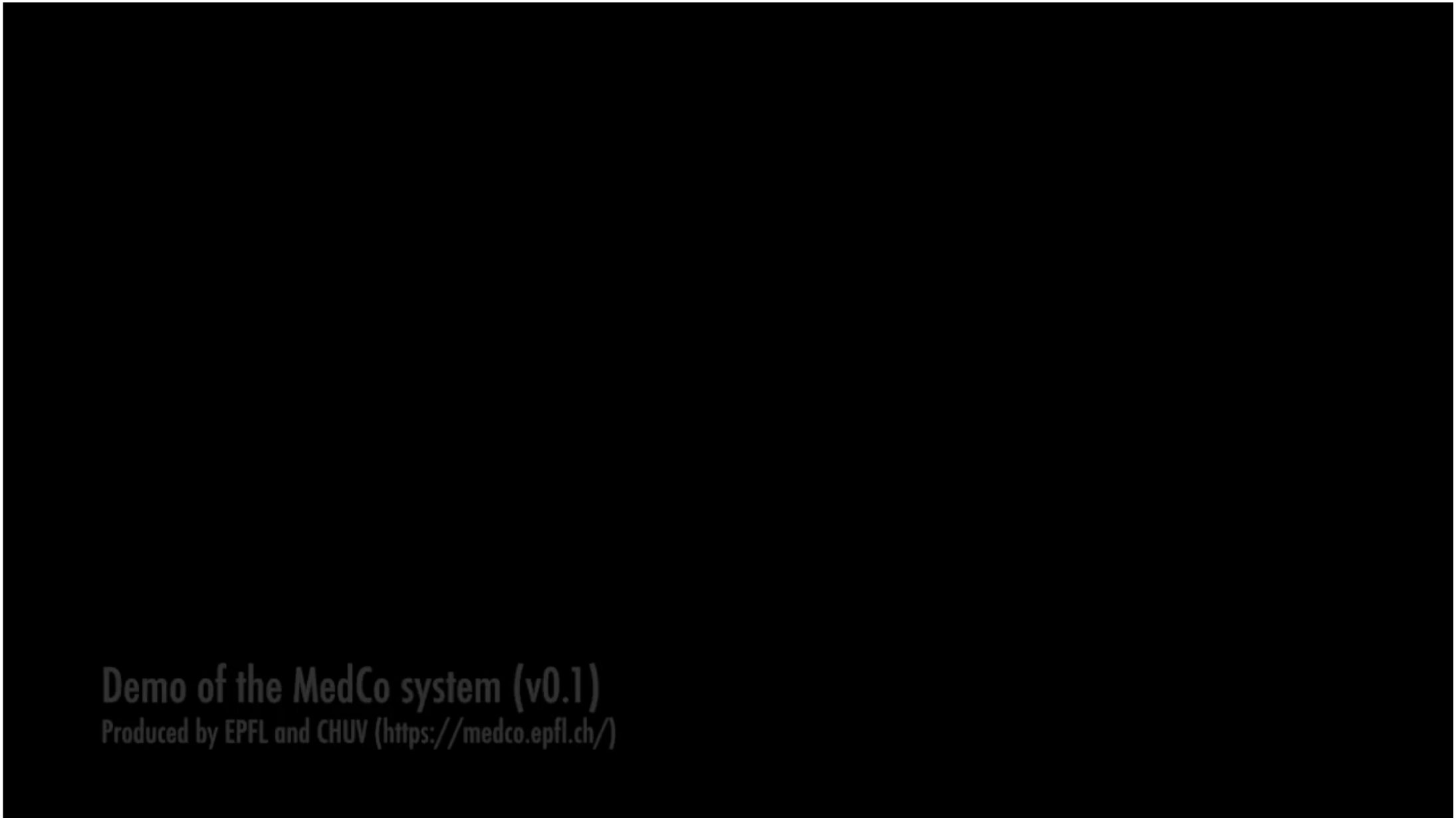
What will happen in the demo



0. Each CS encrypts and stores their data locally

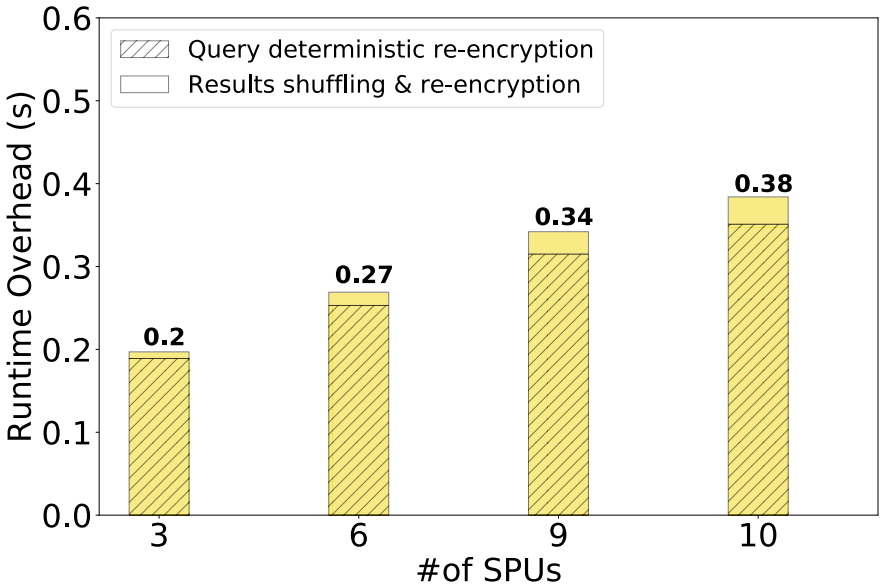
---

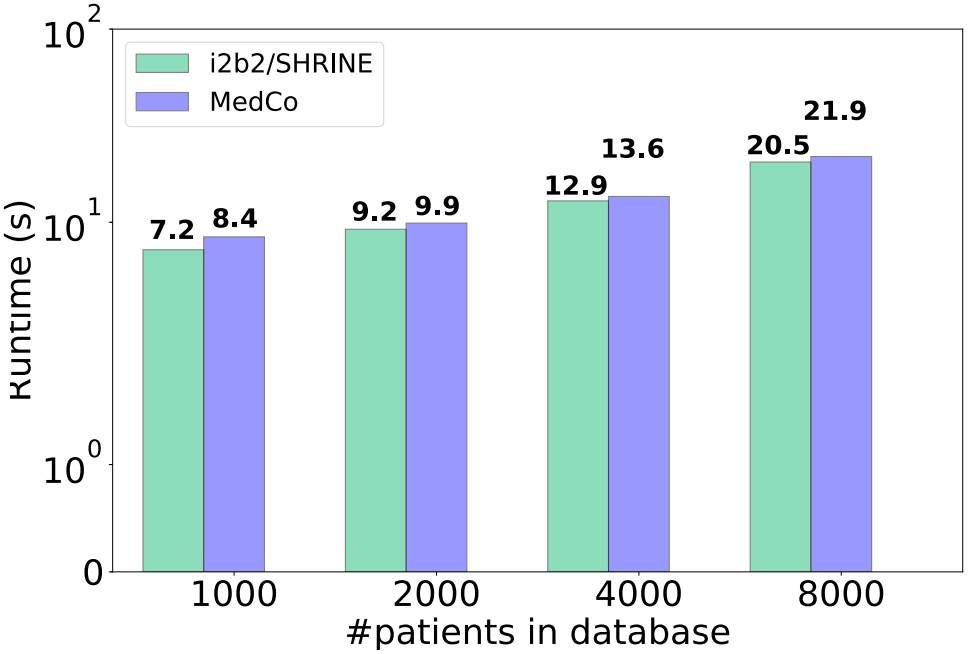
1. The querier fetches the genomic IDs (numeric representation for a genomic variant) that match his/her query.
2. Both the genomic IDs and the clinical attributes (e.g., Cutaneous Melanoma = TRUE) are encrypted.
3. The query is sent to a node (e.g. that can belong to a CS or another entity).
4. The query is broadcast to the other nodes. The encrypted values are matched to the values in each of the CSs' databases (using a series of protocols that are collectively executed by all the nodes).
5. The results, which are still encrypted, are sent to the querier that can decrypt them.


Demo of the MedCo system (v0.1)
Produced by EPFL and CHUV (https://medco.epfl.ch/)

Test on clinical oncology use case

- Experimental Setup:
  - 3 clinical sites with one SPU each
  - 8,000 patients at each site with >1 million clinical and genetic attributes from The Cancer Genome Atlas

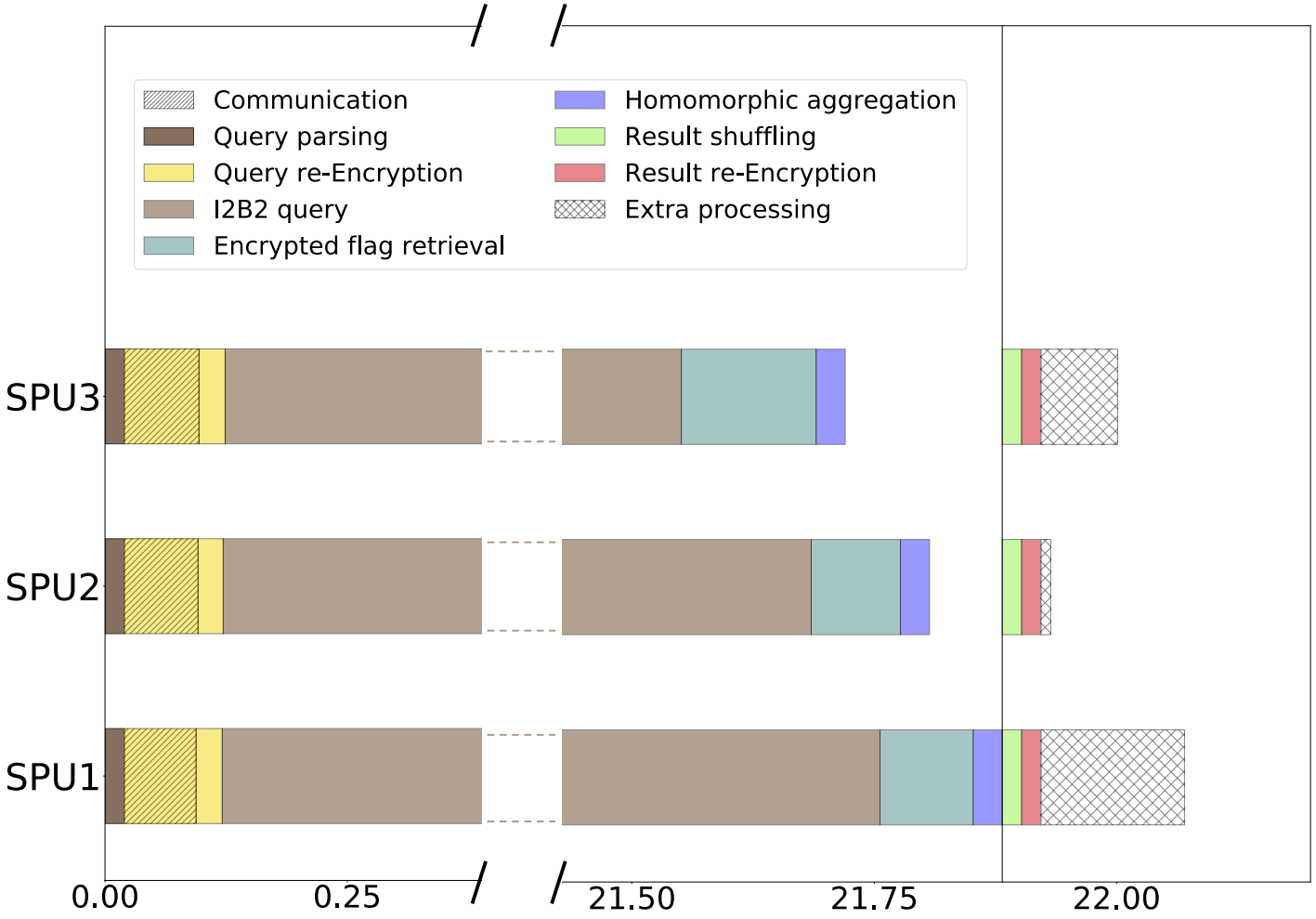




Query: "Number of patients with skin cutaneous melanoma AND a mutation in BRAF gene affecting the protein at position 600."

HW/SW configuration: 3 servers (2.5.GHz Intel Xeon E5-2680 v3 CPU with 12 cores); 256GB of RAM; Postgres SQL; ElGamal on Ed25519 with 128 bits security; UnLynx library (Froelicher et al.'17)

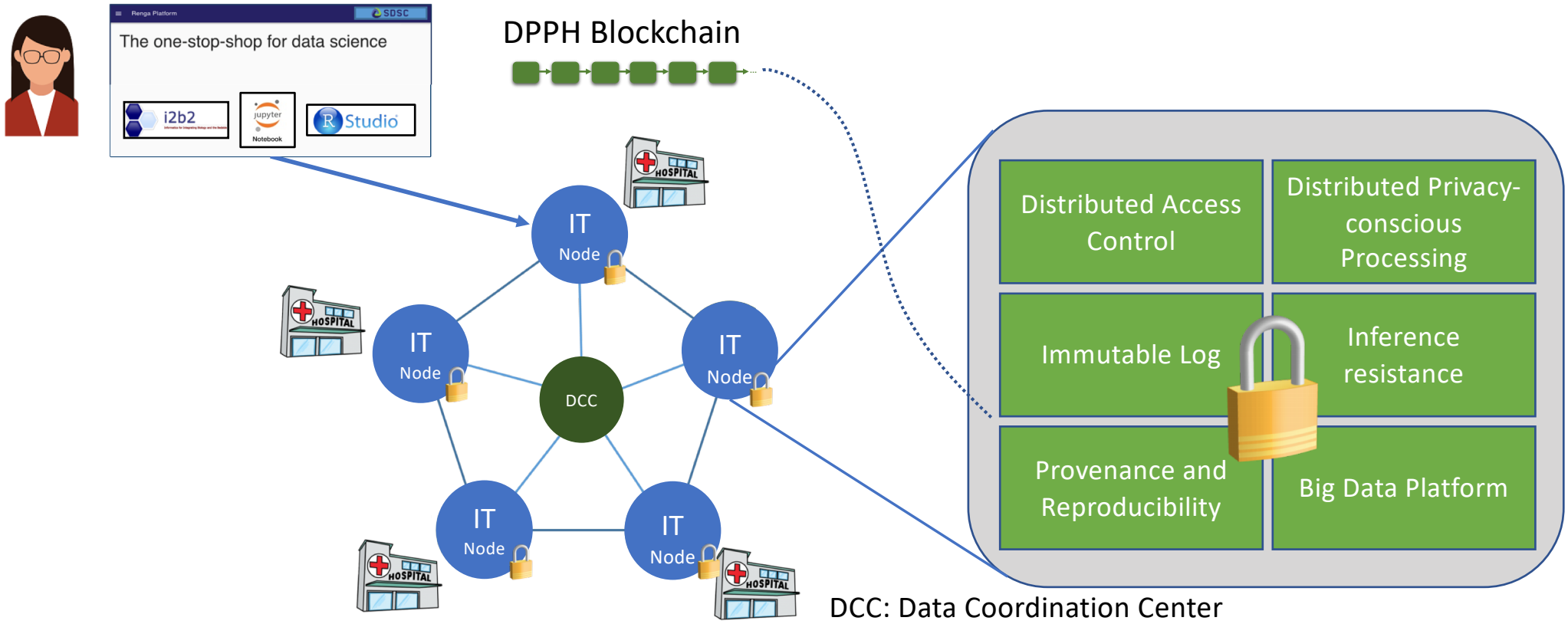
Query runtime break-down



Step 2: Secure and Decentralized Computation on the Selected Records

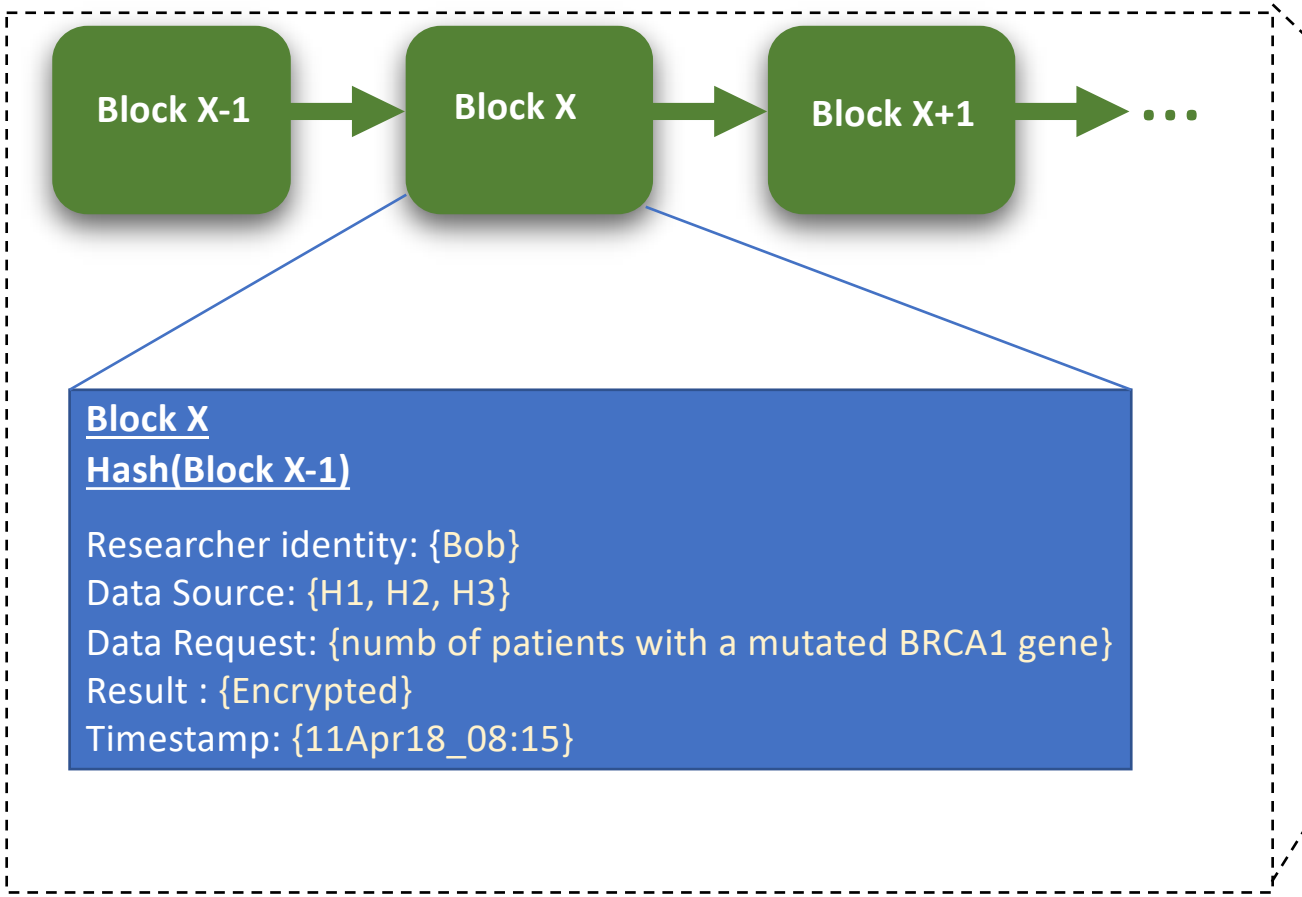
- Features
  - Fully decentralized architecture
  - Data stay with each data provider
  - Resistance against colluding, malicious adversaries
- Linear regression and logistic regression are computable under homomorphic encryption
- Current research: extension to neural networks
- More on this in a few months ☺

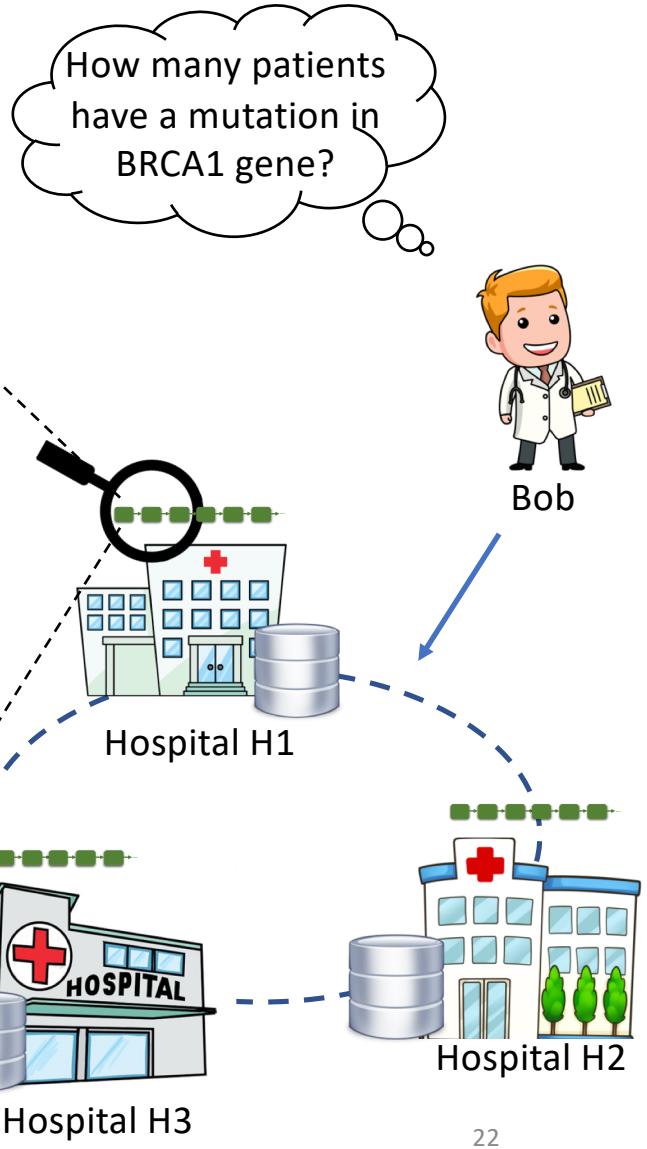
DPPH – The Role of the Blockchain



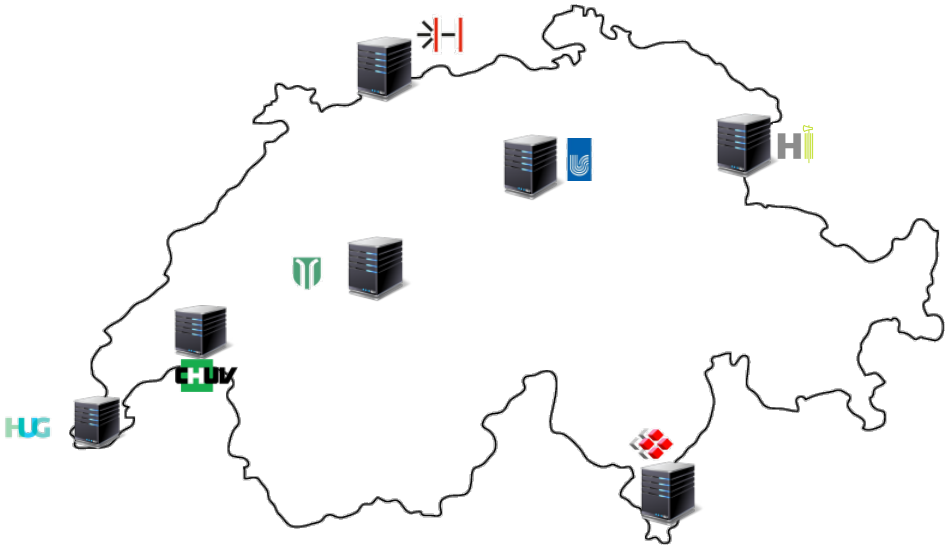
We use a **private** blockchain, unlike Bitcoin that uses a **public** blockchain.

Blockchain as immutable log of data access





Data Protection for Personalized Health



Swiss Personalized Health Network

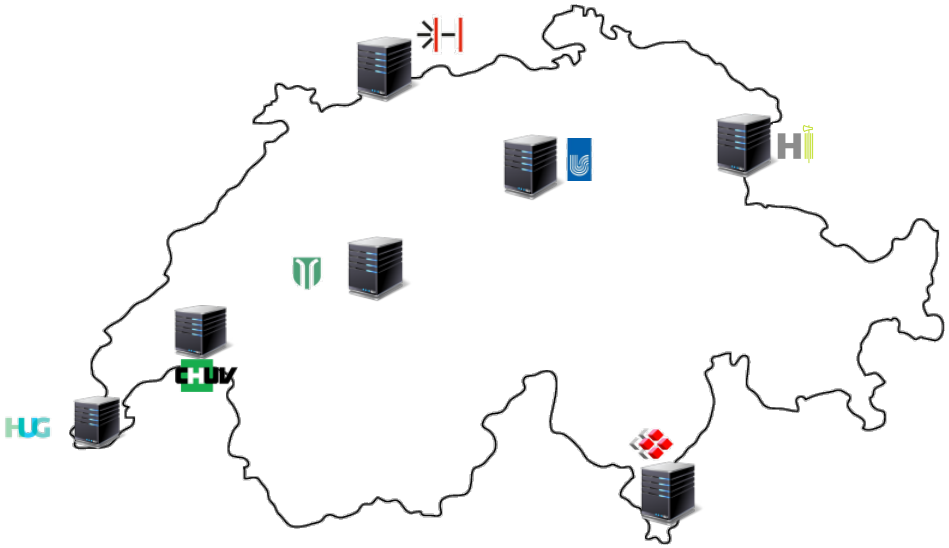
At the international level:

Global Alliance for Genomics & Health

Collaborate. Innovate. Accelerate.

GA4GH has its own workstream on data security

Data Protection for Personalized Health



Swiss Personalized Health Network

At the international level:

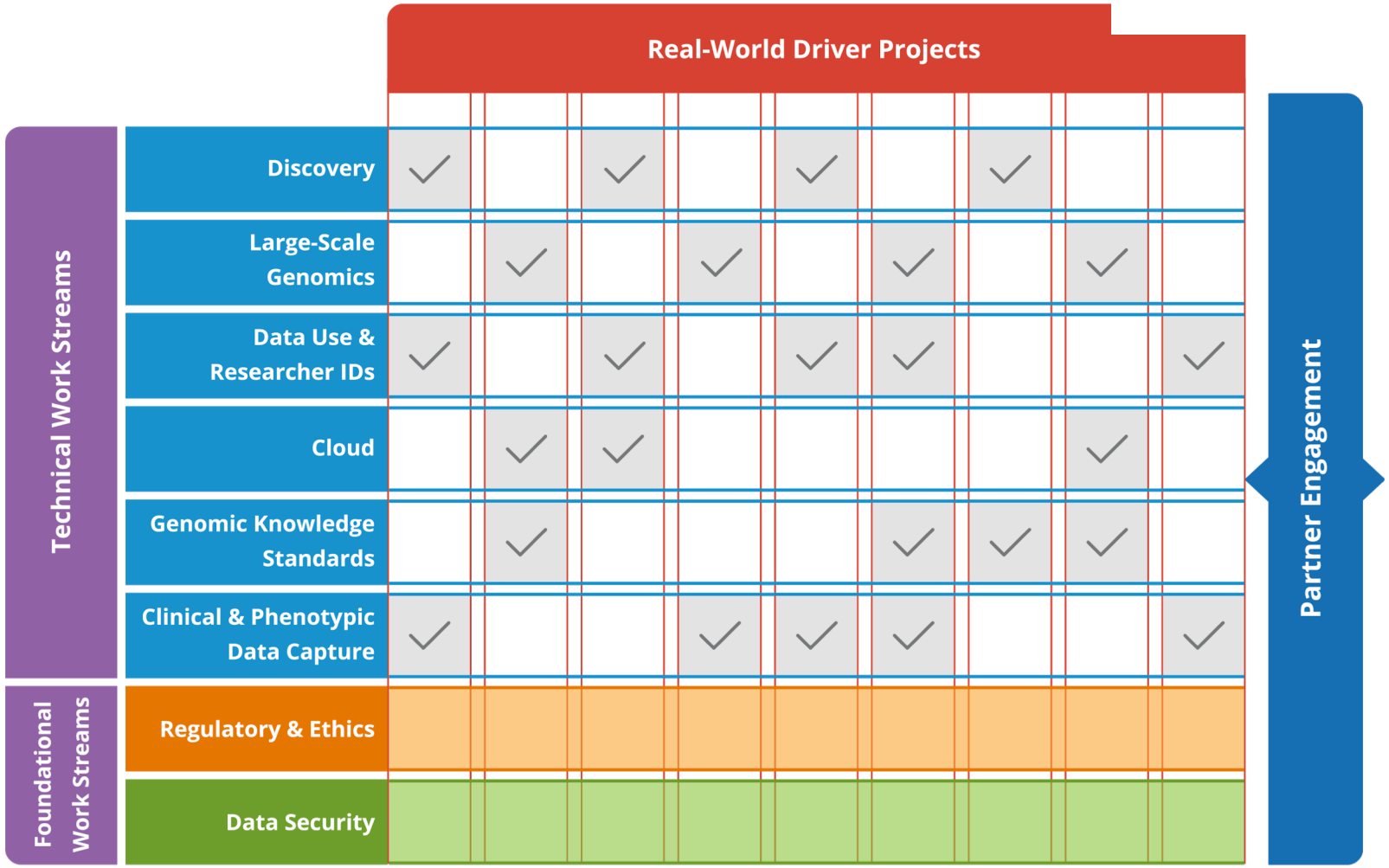
Global Alliance for Genomics & Health

Collaborate. Innovate. Accelerate.

GA4GH has its own workstream on data security

GA4GH Organization

Global Alliance for Genomics & Health
Collaborate. Innovate. Accelerate.



Work Streams vs Driver Projects

Global Alliance for Genomics & Health

Work Streams

- Internal to GA4GH
- Deliver standards and policy frameworks based on the Strategic Roadmap
- Run by 2 volunteer Leads within the community
- Contributors come from a variety of projects and organizations

Example:

Data Use and Researcher Identities

Driver Projects

- External to GA4GH
- Provide input towards the Strategic Roadmap and standards development
- Contribute FTE resources to Work Streams for standards development
- Pilot implementations for new standards

Example:

ARGO Accelerating Research in Genomic Oncology

International Cancer Genome Consortium

Data Security

Global Alliance for Genomics & Health

Technology standards and best practices for protecting data

In development

- **Authentication and authorization infrastructure (AAI)**: GA4GH standard technical profile for authenticating the identity of individuals seeking to access data and services
- **Breach Response Protocol:** protocol for the GA4GH community to effectively respond to and recover from security breaches
- Ongoing discussions
  - with the Cloud Work Stream
  - on homomorphic encryption and SMC

Events on Genome Privacy and Security

- **Dagstuhl** seminars on genome privacy and security 2013, 2015
- **Conference on Genome and Patient Privacy (GaPP)**
  - March 2016, Stanford School of Medicine
- **GenoPri**: International Workshop on Genome Privacy and Security
  - July 2014: Amsterdam (co-located with PETS)
  - May 2015: San Jose (co-located with IEEE S&P)
  - November 12, 2016: Chicago (co-located with AMIA)
  - October 15, 2017: Orlando (co-located with Am. Society for Human Genetics (ASHG) and GA4GH)
  - October 3, 2018, Basel (co-located with GA4GH)
  - October 21-22, 2019, Boston (co-located with GA4GH)

SCHLOSS DAGSTUHL
Leibniz-Zentrum für Informatik

- **iDash**: integrating Data for Analysis, Anonymization and sHaring (already in previous years)
  - **October 14, 2017: Orlando**

iDASH

- Inst. For Pure and Applied Mathematics (IPAM, UCLA)

  Algorithmic Challenges in Protecting Privacy for Biomed Data

  10-12 January, 2018

ipam

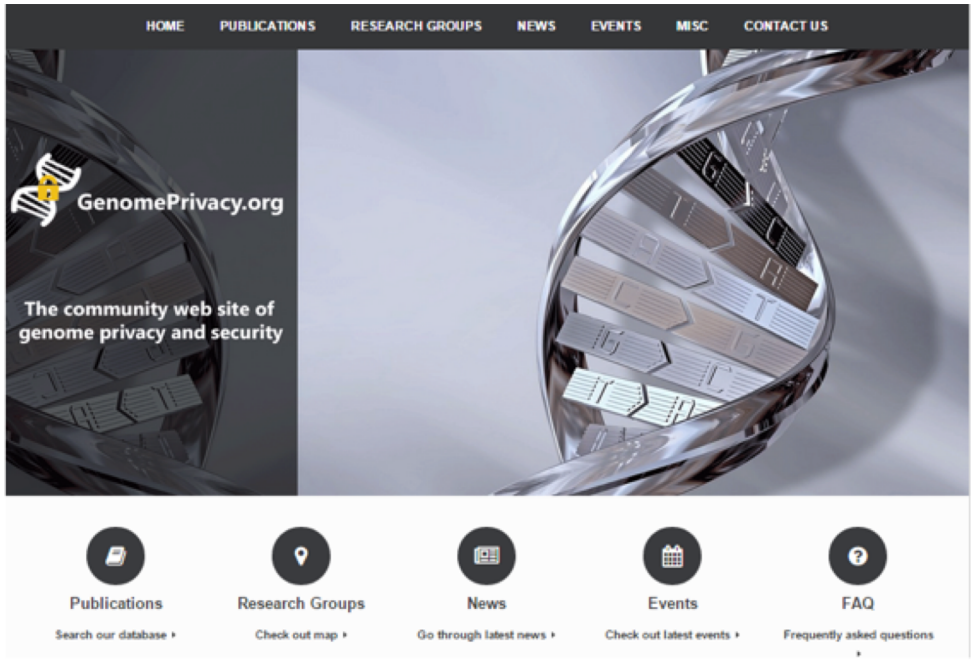
- DPPH Workshop, 15 February 2018

DPPH18

➔ Lots of material online

DPPH18.epfl.ch

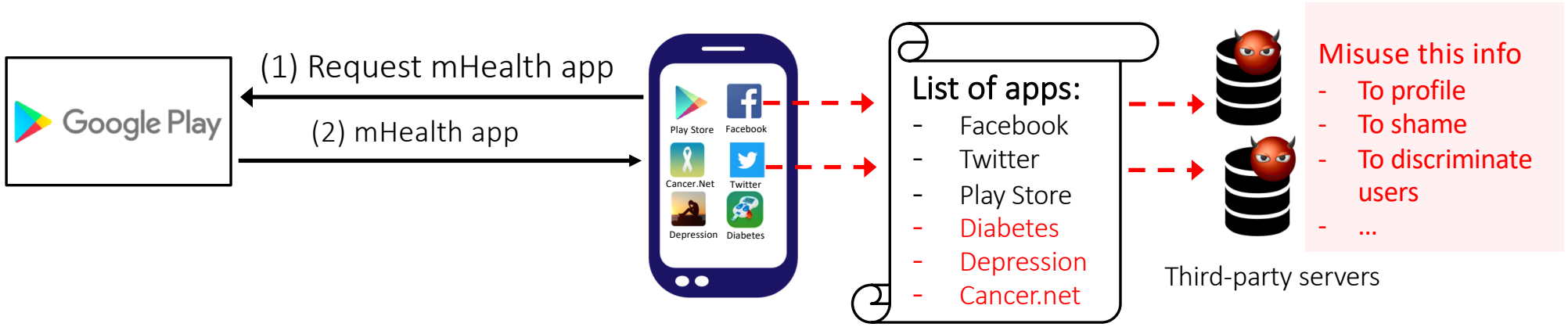
“genomeprivacy.org”



Community website

- Searchable list of publications on genome privacy and security
- News from major media (from Science, Nature, GenomeWeb, etc.)
- Research groups and companies involved
- Tutorial and tools
- Events (past & future)

Privacy Challenge in mHealth

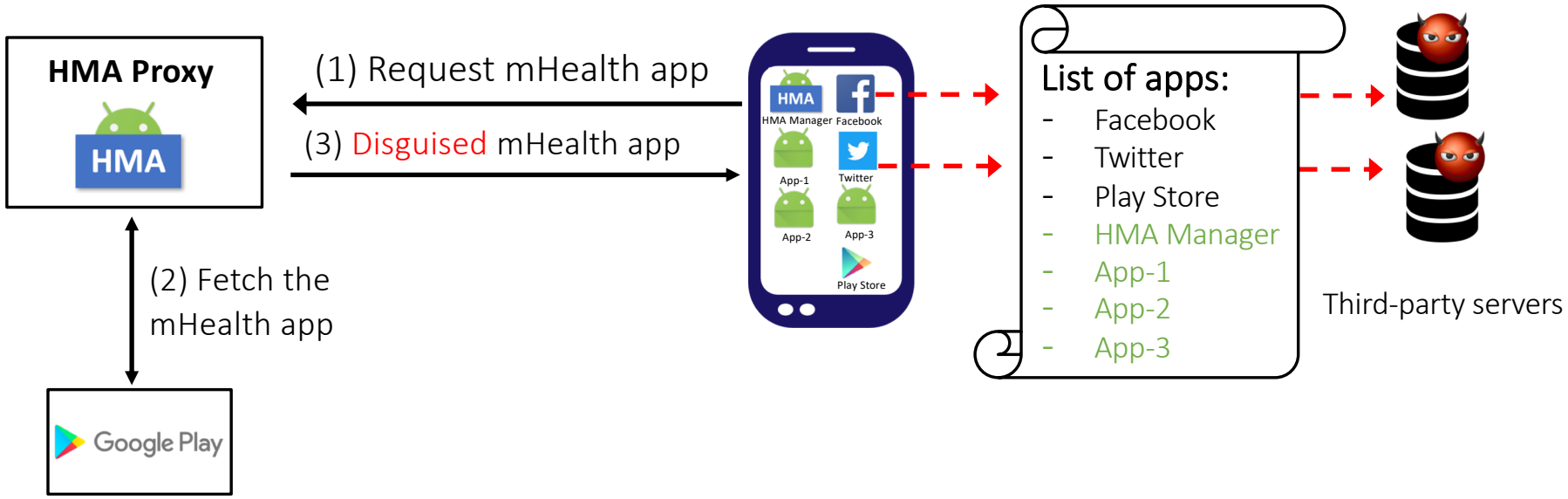


- Many apps collect the list of installed apps
- Presence of an mHealth app → specific medical conditions of its users
- Collected lists of installed apps can be shared with third-parties

How to hide the presence of a sensitive app from other apps while preserving key functionalities and usability of the app, and without requiring users to modify the OS of their phones?

Solution: HideMyApp (HMA)

- Main idea: Launch the sensitive app without installing it



- Technologies used:
  - Dynamic loading of classes and resources from an application package (APK)
  - App virtualization
  - Randomization and obfuscation

Conclusion

- Protecting health data is one of the most formidable challenges for cybersecurity
- With the advent of genomics:
  - risk is increasing
  - conventional medical data protection techniques based on de-identification do not work anymore
- There exist technical solutions to address the problem
- Legislation will also play an important role