



Children's Hospital Colorado



An Infrastructure for Secure Sharing of Clinical Data

Joshua Roberts and David Ferraiolo
National Institute of Standards and Technology



Analytics with Altitude

2023 HDAA Conference

Denver, Colorado

AGENDA

- Overview: Secure Federated Data Sharing System (SFDS)
- SFDS value in performing clinical research
- Demonstration



SFDS Overview



Analytics with Altitude **2023 HDAA Conference** *Denver, Colorado*



Children's Hospital
Colorado



Problem

How can organizations securely share data?

- The ability to share database resources among collaborating organizations is highly desirable – this is especially true in the performance of clinical research.
- However, challenges persist regarding interoperability in the exchange of resources among organizations and preservation of organization's distinct protection policies.
- Hard for users in different organizations to share DBMS data. Because the data is
 - from different systems,
 - in different formats,
 - organized under different schemas
 - protected under the host's access control policies.

Solution

Exchange attributes not data

- A standard means of providing policy-preserving access to the data where it currently resides, rather than its exchange, or centralized storage.
- Transparent to the otherwise normal business operations of participating organizations.
- Accomplished using two NIST developed technologies
 - **Data Block Matrix (DBM)** – Verify user's attributes across a federation of organizations
 - **Next generation Database Access Control (NDAC)** – Control access to SQL databases with cell level access control
- Through consent, previously unknown users are onboarded into local **NDAC** systems using their **DBM** validated attributes, allowing them policy preserving access to local database resources.

Data Block Matrix (DBM)

A permissioned distributed ledger with the ability to edit or delete data.

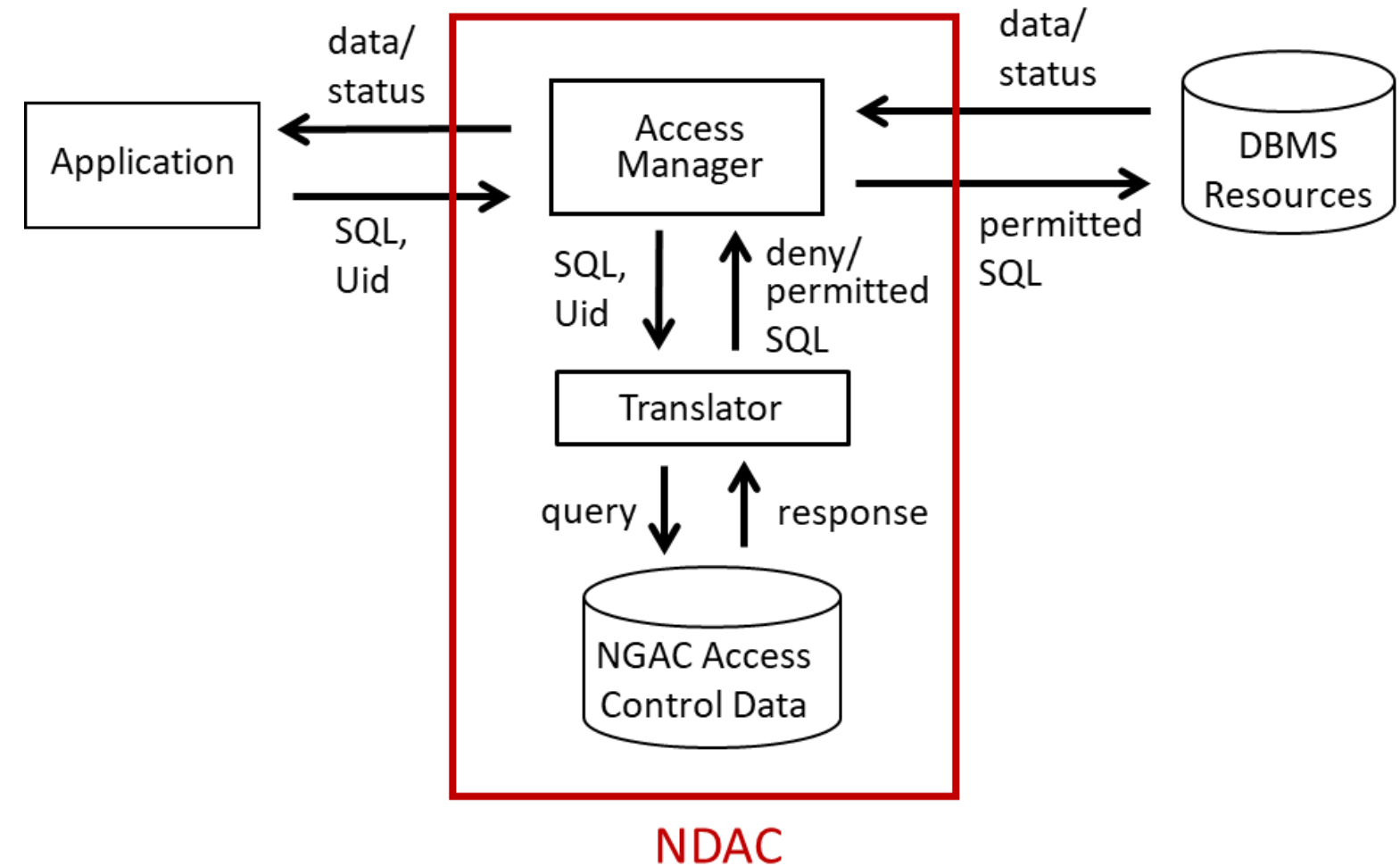
- A permissioned DLT, with the hashed data integrity protection of a blockchain, but with the additional ability to **edit or delete data**.
- Manage and verify user attributes for all federation users.
- A **standard catalog** of attributes agreed upon by the federation is used in their local **NGAC** policies (i.e. SNOMED).
- NGAC governance policy controls who can update user attribute assignments.

	0	1	2	3	4	
0	•	1	3	7	13	$H_{0,-}$
1	2	•	5	9	15	$H_{1,-}$
2	4	6	•	11	17	$H_{2,-}$
3	8	10	12	•	19	$H_{3,-}$
4	14	16	18	20	•	$H_{4,-}$
	$H_{-,0}$	$H_{-,1}$	$H_{-,2}$	$H_{-,3}$	$H_{-,4}$	etc.

Editing or deleting block **12** only requires the update of row hash $H_{3,-}$ and column hash $H_{-,2}$

Next generation Database Access Control (NDAC)

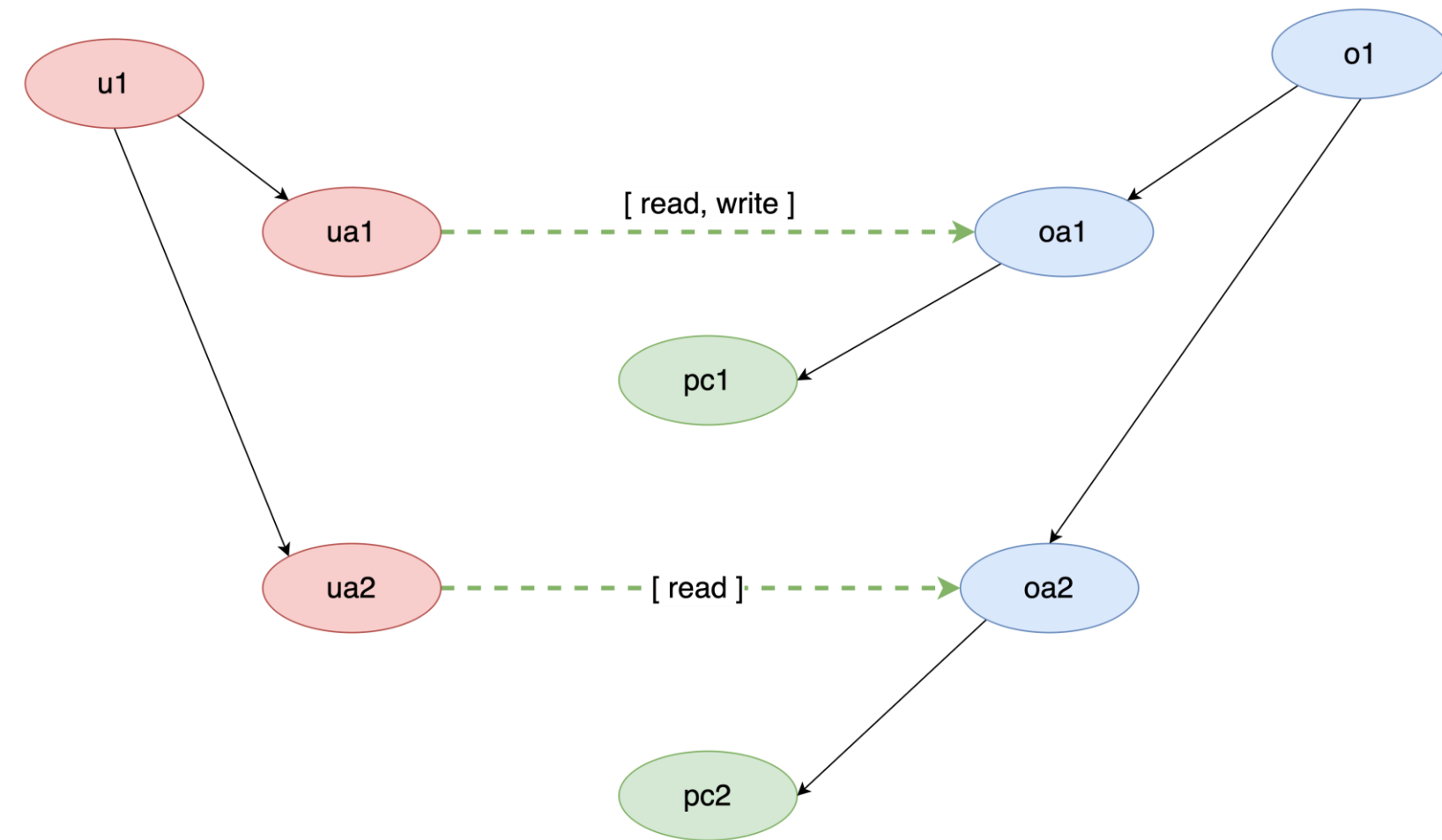
- Uses **Next Generation Access Control (NGAC)** to **translate** SQL statements into permitted SQL.
- DBMS and application agnostic.
- Enforce policies at a granularity not typically available to DBMSs.
- NGAC policy review allows access state to be cached, speeding up translation at query time.
- Performance: ~3 seconds per 1 million rows when caching.



Next Generation Access Control (NGAC)

ANSI/INCITS ABAC Standard

- **Privileges** are represented as a Directed Acyclic Graph (DAG).
 - Based on user attribute assignments, resource (objects) attribute assignments, and other relations specified in terms of those attributes.
- **Prohibitions** supplement policy graph and deny permissions for users on objects.
- **Obligations** dynamically change access state in response to policy and external events.



User **u1** has **[read]** access rights on **o1**

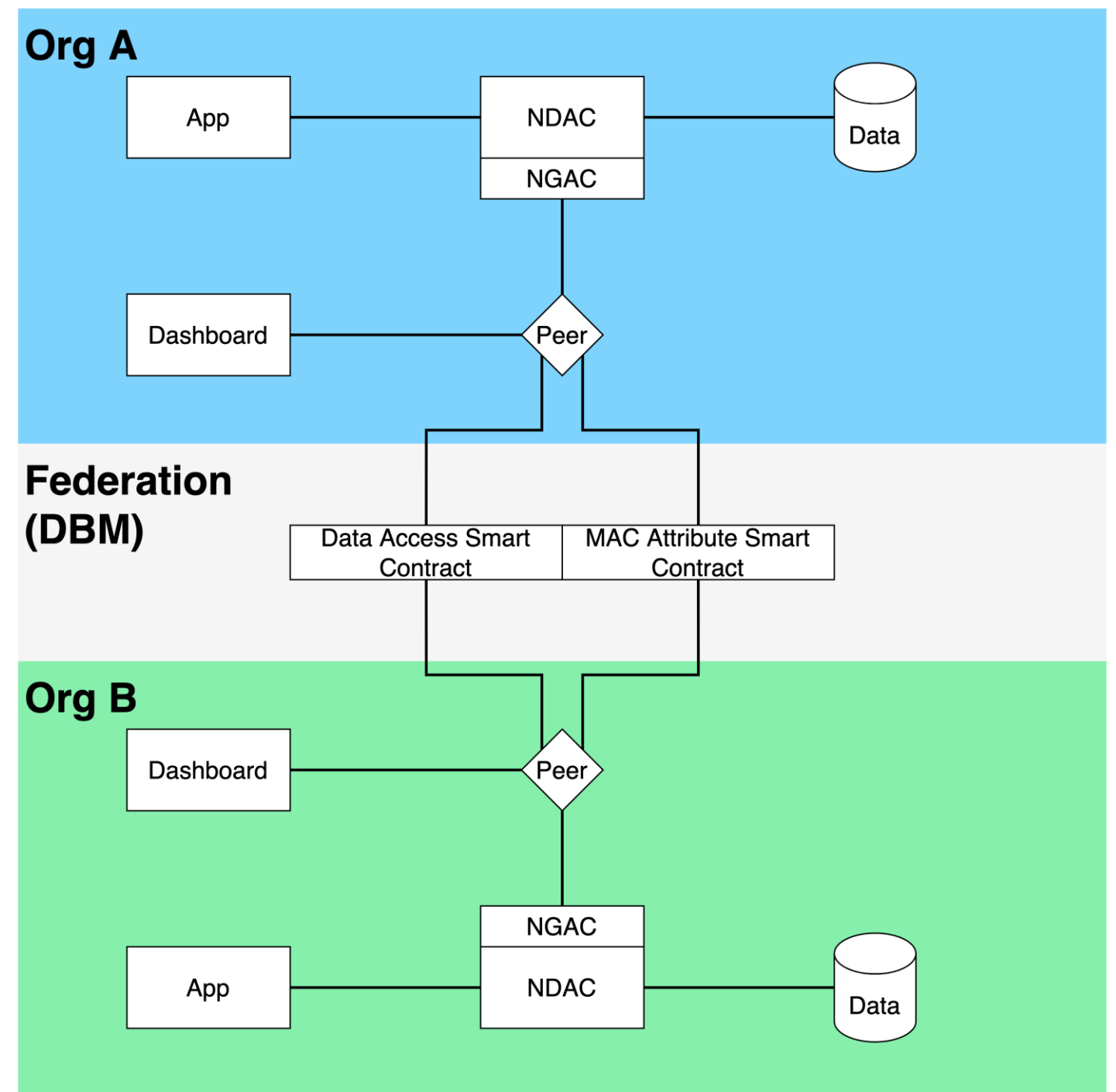


Access Control Policies

- **Discretionary Access Control (DAC)** - Provide organizations with capabilities to grant or prohibit DBM users access to their data sets.
- **Mandatory Access Control (MAC)** - Policies impose non-discretionary rules on users when accessing resources for the enforcement of mandated policies or regulation based on roles or other types of attributes.
- NGAC enforces policy combinations.

SFDS Architecture (2 Orgs)

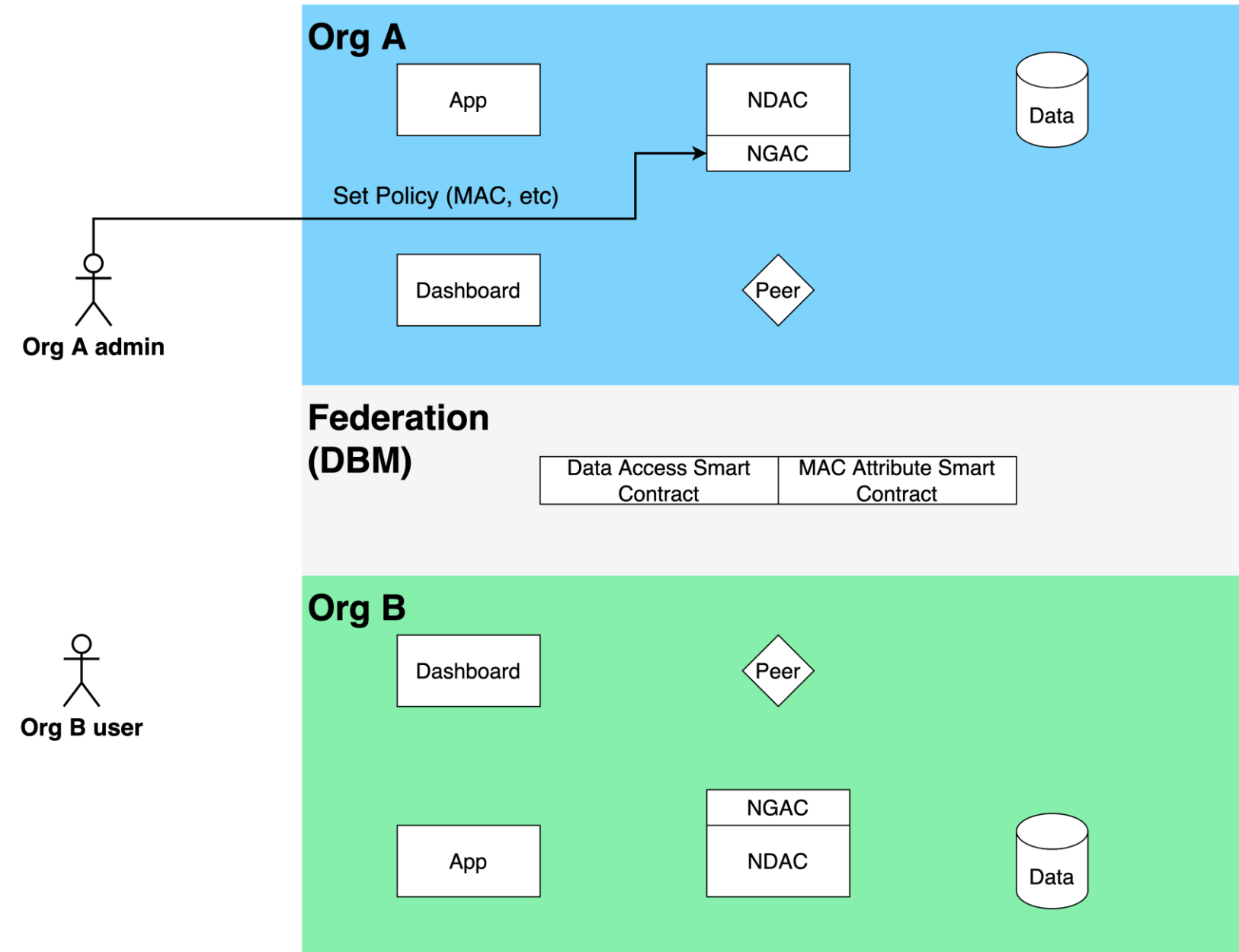
- Data access application stack (**App**, **Data**)
- An **NDAC** server with **NGAC** policy to protect the data
- **Dashboard** to request/grant access to local resources and **Peer** node to interact with DBM smart contracts



1. Set initial NGAC policy

Scenario: Org B user accesses Org A data

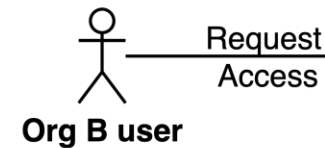
- Org A admin sets initial NGAC policy using, but not limited to, the attributes defined in the standard catalog



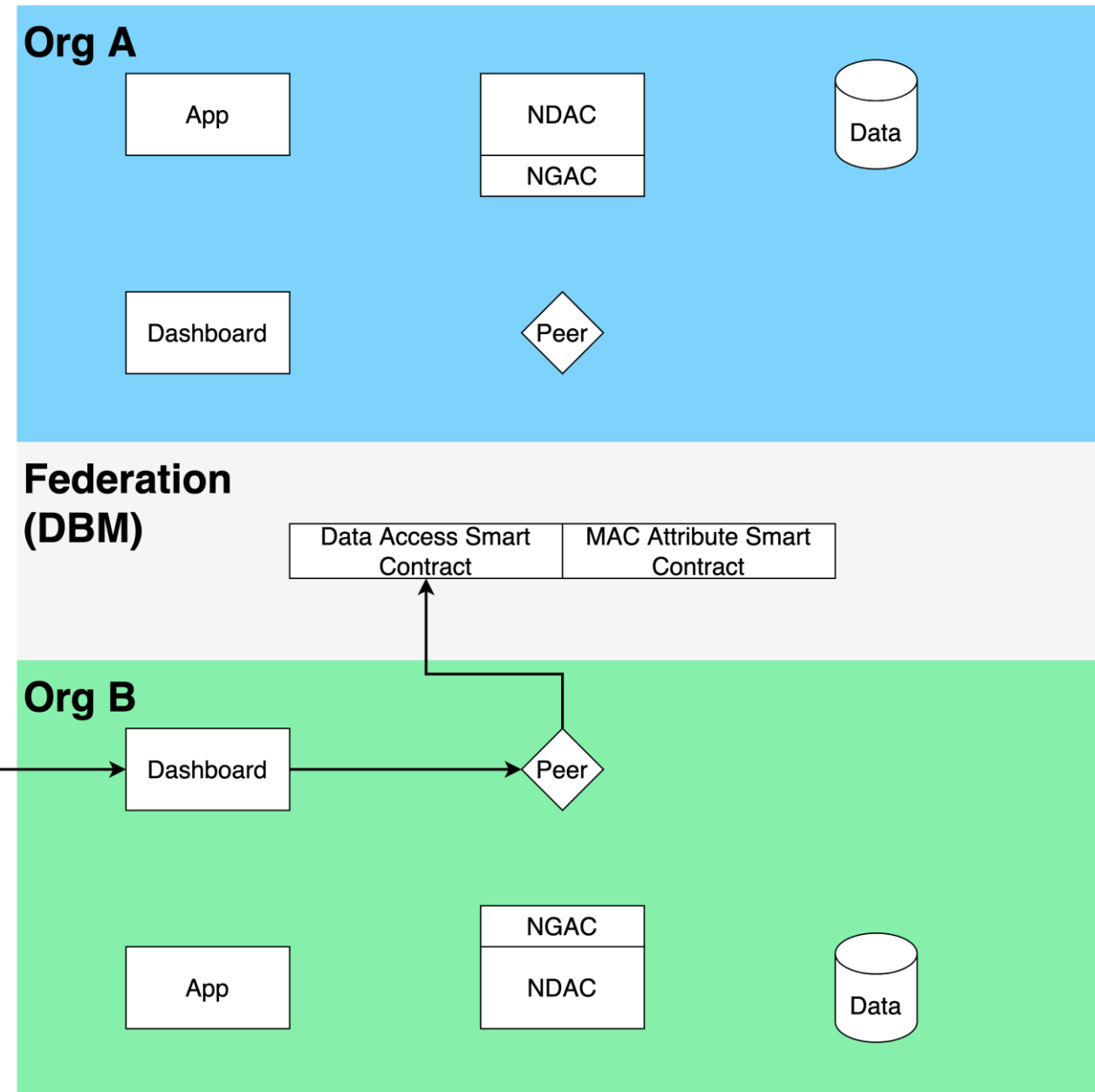
2. Request access

Scenario: Org B user accesses Org A data

- Org B user requests access to Org A's data using the Org B dashboard which invokes the Data Access Smart Contract



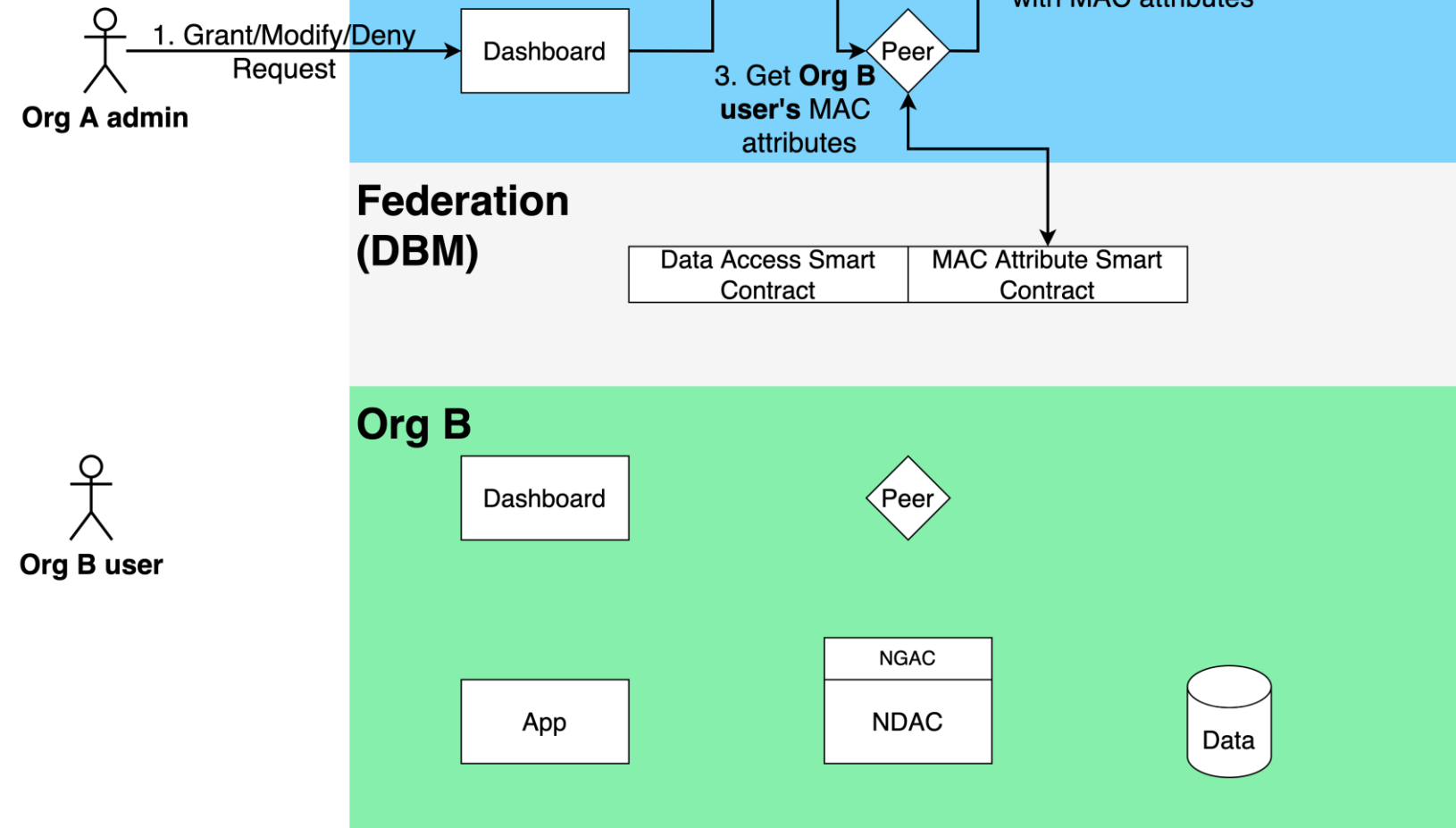
Request
Access



3. Set DAC policy and onboard user

Scenario: Org B user accesses Org A data

1. Org A admin grants/modifies the request in Org A's dashboard.
2. Org A admin sends the required **DAC** policy to NGAC.
3. NGAC retrieves the MAC attributes from the MAC Attribute Smart Contract.
4. NGAC assigns the user to the matching **MAC** attributes in the local policy graph.



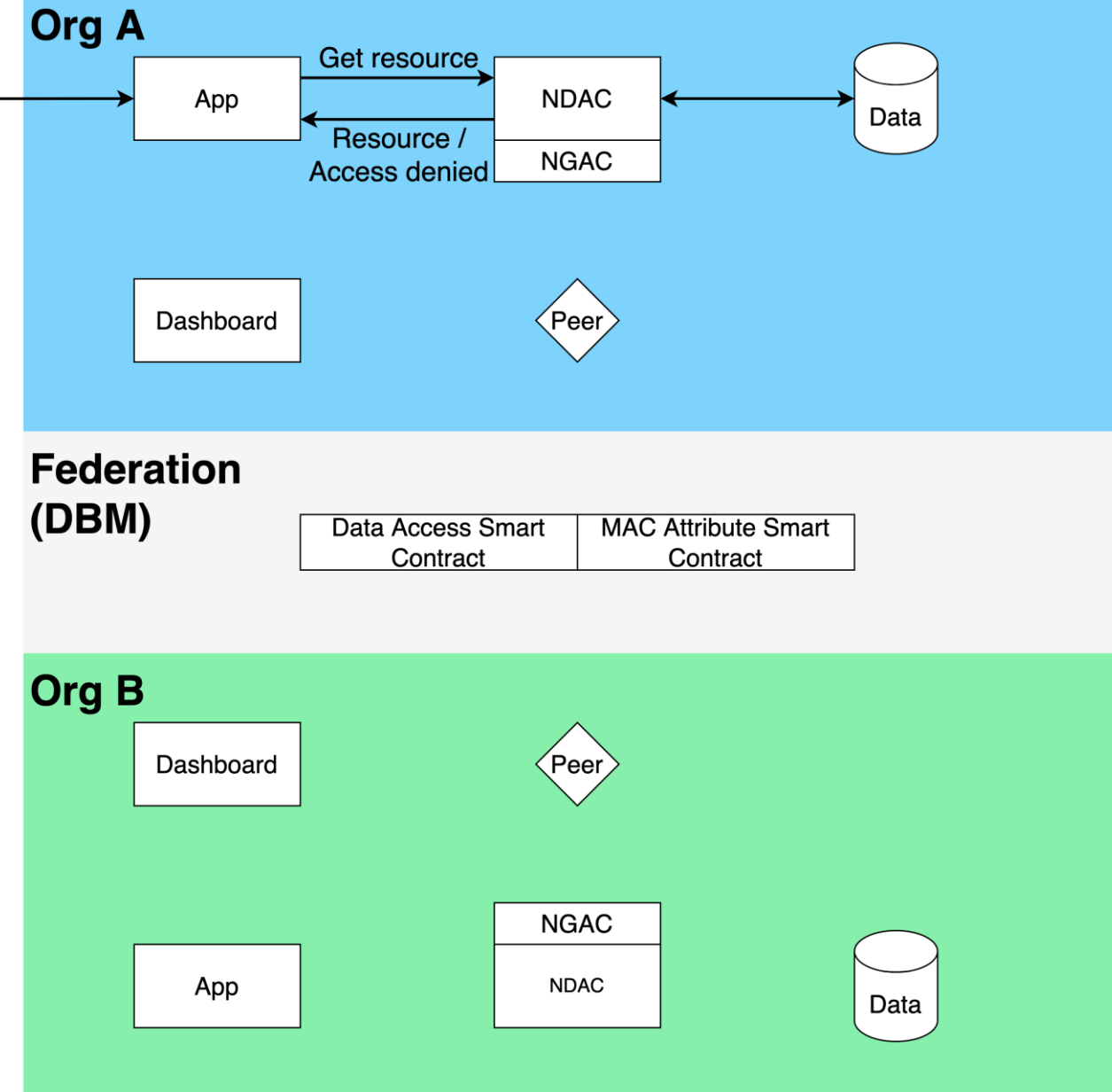
4. Access Data

Scenario: Org B user accesses Org A data

- Org B user accesses Org A's data the same as a regular Org A user
- Organizations have control over the authentication of external users that access their data

Org A admin

Org B user



SFDS value in performing
clinical research



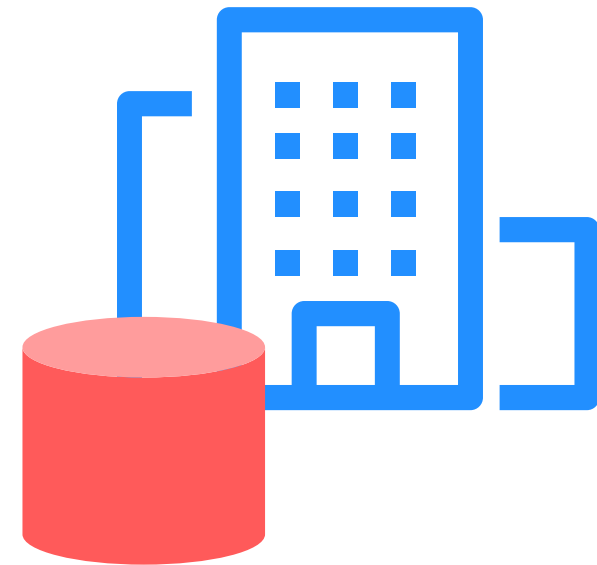
Analytics with Altitude
2023 HDAA Conference
Denver, Colorado



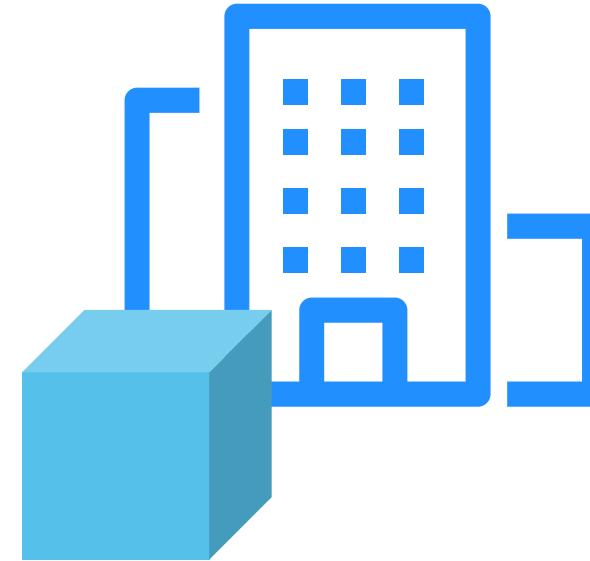
Children's Hospital
Colorado



Clinical Data Sharing Challenge



Institution A
Has a very large dataset
with **few restrictions**.



Institution B
Has a **more restrictive posture** with sharing
due to the nature of
some of the population
served.



Institution C
Has significant **research expertise** and resources.



Analytics with Altitude
2023 HDAA Conference
Denver, Colorado



The Participants

- Technologists
- Administrators
- Researchers
- Clinicians
- Subject Matter Experts
- Governance Bodies
- Assistants/Support
- Students



Likely to Vary Across Institutions



Some May Serve Multiple Roles



Participants May Change Over Time



Barriers to Data Sharing

- Potential misuse or misinterpretation of data
- Lack of resources
- Loss of control
- Socio-cultural factors and ethical and legal barriers

Devriendt T, Borry P, Shabani M (2021)

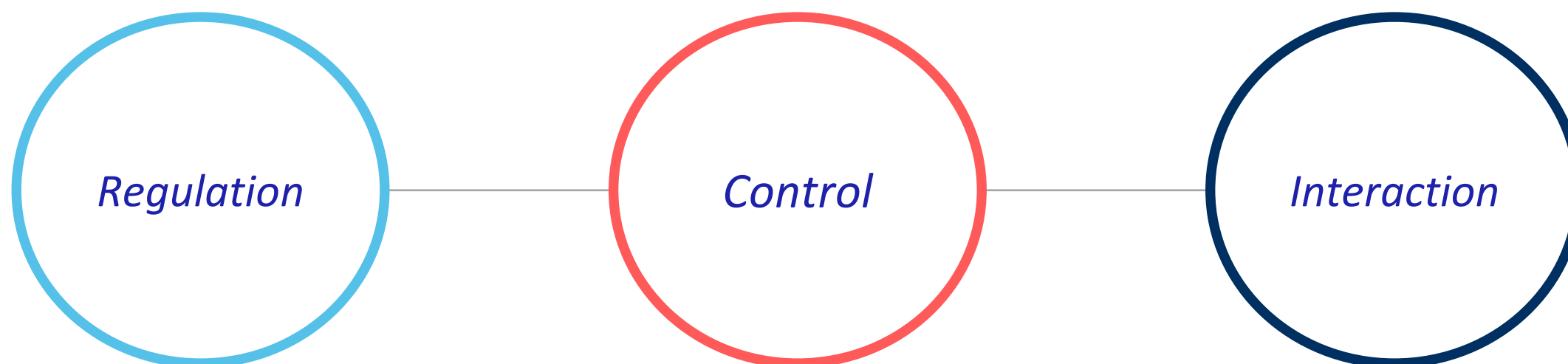
Factors that influence data sharing through data sharing platforms: A qualitative study on the views and experiences of cohort holders and platform developers.



Analytics with Altitude
2023 HDAA Conference
Denver, Colorado



Data Sharing Questions



**What are we required to
do?**

Are we sure it is protected?

**How do we provide
access?**



Analytics with Altitude
2023 HDAA Conference
Denver, Colorado



Demonstration



Analytics with Altitude **2023 HDAA Conference** *Denver, Colorado*

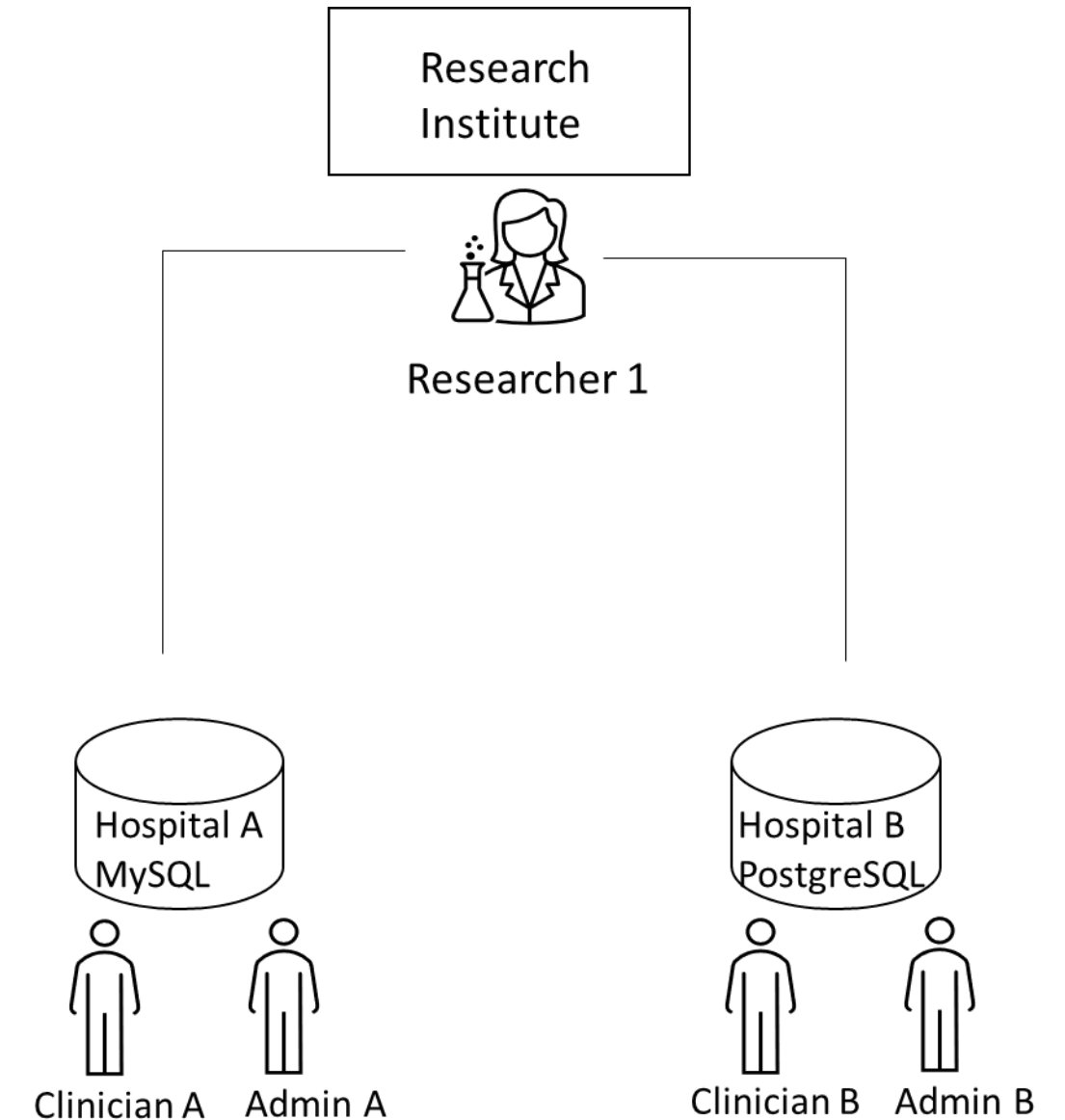


Children's Hospital
Colorado



Use Case Scenario

- **Researcher 1** wants access to clinical data at **Hospital A** and **Hospital B** to perform an analysis of the variables and indicators of patients at risk of **sepsis**.
- Hospital A and Hospital B restrict access to patient **PII** and data related to **mental health** (MAC policy).
- Hospital B considers substance abuse related to mental health, while Hospital A does not.
- Both hospitals will create similar DAC policies which grant Researcher 1 access to **patient identity** and **financial information** for only sepsis patients.



1. Researcher1 requests access to data at HospitalA and HospitalB

Research Institute Access Dashboard

?

Requests

☰

Policies

🔍

Search

👤

researcher1

Requests

+

Requester

researcher1

Target Organization

Organizations

HospitalA

Request

sepsis research

Submit Request

1. Researcher1 requests access to data at HospitalA and HospitalB (cont)

Research Institute Access Dashboard

?

Requests

:

Policies

Q

Search

researcher1

Requests

+

Requester

researcher1

Target Org

HospitalA

Requester

researcher1

Target Org

HospitalB

2. HospitalA and HospitalB admins set DAC policies for Researcher1

HospitalA Access Dashboard? Requests≡ Policies🔍 Search👤 a_admin

Requests+

Requester

researcher1

Target Organization

Organizations
HospitalA

Request

sepsis research

Discretionary Access Control (DAC) Policy

```
let username          = "researcher1";
let permissions        = ["select", "insert"];
let policyUA           = concat([username, " policy-UA"]);
let policyOA           = concat([username, " policy-OA"]);
let patientsSubPolicyOA = concat([username, " patients sub-policy"]);
let encountersSubPolicyOA = concat([username, " encounters sub-policy"]);
let proceduresSubPolicyOA = concat([username, " procedures sub-policy"]);

if nodeExists(policyOA) {
  deassignAll(getChildren(policyUA), policyUA);
  deassignAll(getChildren(patientsSubPolicyOA), policyOA);
  deassignAll(getChildren(encountersSubPolicyOA), policyOA);
  deassignAll(getChildren(proceduresSubPolicyOA), policyOA);
} else {
  create oa policyOA in ["DAC"];
  create ua policyUA in ["DAC"];
  create oa patientsSubPolicyOA in [policyOA];
  create oa encountersSubPolicyOA in [policyOA];
  create oa proceduresSubPolicyOA in [policyOA];
}

assign username to [policyUA].
```

Set Policy

Load Demo DAC

Revoke

HospitalB Access Dashboard? Requests≡ Policies🔍 Search👤 b_admin

Requests+

Requester

researcher1

Target Organization

Organizations
HospitalB

Request

sepsis research

Discretionary Access Control (DAC) Policy

```
let username          = "researcher1";
let permissions        = ["select", "insert"];
let policyUA           = concat([username, " policy-UA"]);
let policyOA           = concat([username, " policy-OA"]);
let patientsSubPolicyOA = concat([username, " patients sub-policy"]);
let visitsSubPolicyOA   = concat([username, " visits sub-policy"]);
let treatmentsSubPolicyOA = concat([username, " treatments sub-policy"]);
let connUrl            = "jdbc:postgresql://host.docker.internal:5432/B?user=postgres&password=root";

if nodeExists(policyOA) {
  deassignAll(getChildren(policyUA), policyUA);
  deassignAll(getChildren(patientsSubPolicyOA), policyOA);
  deassignAll(getChildren(visitsSubPolicyOA), policyOA);
  deassignAll(getChildren(treatmentsSubPolicyOA), policyOA);
} else {
  create oa policyOA in ["DAC"];
  create ua policyUA in ["DAC"];
  create oa patientsSubPolicyOA in [policyOA];
  create oa visitsSubPolicyOA in [policyOA];
  create oa treatmentsSubPolicyOA in [policyOA];
}
```

Set Policy

Load Demo DAC

Revoke

3. Researcher1 reviews DAC policies and obtains authentication tokens

Research Institute Access Dashboard

?

Requests

≡

Policies

🔍

Search

👤

researcher1

Policies

Requester
researcher1
Target Org
HospitalA

Requester
researcher1
Target Org
HospitalB

Requester

researcher1

Password

cpms5zqu

Target Organization

Organizations

HospitalA

Request

sepsis research

Discretionary Access Control (DAC) Policy

```
let username = "researcher1";
let permissions = ["select", "insert"];
let policyUA = concat([username, " policy-UA"]);
let policyOA = concat([username, " policy-OA"]);
let patientsSubPolicyOA = concat([username, " patients sub-policy"]);
let encountersSubPolicyOA = concat([username, " encounters sub-policy"]);
let proceduresSubPolicyOA = concat([username, " procedures sub-policy"]);

if nodeExists(policyOA) {
  deassignAll(getChildren(policyUA), policyUA);
  deassignAll(getChildren(patientsSubPolicyOA), policyOA);
  deassignAll(getChildren(encountersSubPolicyOA), policyOA);
  deassignAll(getChildren(proceduresSubPolicyOA), policyOA);
} else {
  create oa policyOA in ["DAC"];
  create ua policyUA in ["DAC"];
  create oa patientsSubPolicyOA in [policyOA];
  create oa encountersSubPolicyOA in [policyOA];
  create oa proceduresSubPolicyOA in [policyOA];
}

assign username to [policyUA];
```

4. A_clinician selects from patients table at HospitalA

A

Filter

ENTITIES 9

> conditions

> encounters

> medications

> observations

> organizations

> patients

> payers

> procedures

> providers

A Clinician - A - Beekeeper Studio

select patients

select procedures (mysql)

1 select * from patients;

Save

Run

id	birthdate	deathdate	ssn	drivers	passport
006a412e-23bd-97d6-8006-a815d0d7f092	1993-01-12	2022-07-24	999-58-8900	S99960852	X20069926X
0188428a-1a3c-134f-0531-979b7c9370bc	1976-09-10	(EMPTY)	999-32-1037	S99944782	X31730780X
02e60b9e-4dac-0e17-8fca-168614d621f7	1956-05-15	(EMPTY)	999-26-8429	S99975377	X27984672X
0648a763-d3a6-b9ce-4073-efc1a610e524	1956-12-14	(EMPTY)	999-55-8998	S99925484	X34324929X
066887db-ae85-9197-4888-c4b57b25422d	1961-07-05	(EMPTY)	999-94-8214	S99924827	X19887271X
0dbceeb8-74af-c0ed-138a-d12e51ea6fe2	2003-08-21	(EMPTY)	999-12-3736	S99942429	(EMPTY)
0efb7080-cc74-1b15-6ec1-fb5ee6e0459e	1972-05-10	(EMPTY)	999-64-9095	S99972500	X41659182X
0fe74a27-9f69-5b08-c2bc-f967b38e3811	2020-02-05	(EMPTY)	999-78-6960	(EMPTY)	(EMPTY)
115722a9-0af5-58f7-eafa-b7e8750929d9	2017-10-09	(EMPTY)	999-89-6816	(EMPTY)	(EMPTY)
120eeb2b-b1e2-2698-01e1-7a9d60d520f0	1997-07-12	(EMPTY)	999-90-7420	S99906042	X12190070X

A Clinician - A

mysql

118 0 affected 0.148 s

Download

5. A_clinician selects from procedures table at HospitalA

A

Filter

ENTITIES 9

conditions

encounters

medications

observations

organizations

patients

payers

procedures

providers

A Clinician - A - Beekeeper Studio

select patients

select procedures (mysql)

```
1 select
2   count(*) as count,
3   reasoncode,
4   reasondescription,
5   round(sum(procedures.base_cost), 2)
6 from procedures
7 where patient in (select patient from encounters where reasoncode = '91302008')
8 group by reasoncode, reasondescription
9 order by count desc;
```

Save

Run

count	reasoncode	reasondescription	round(sum(procedures.bas
2421	(EMPTY)	(EMPTY)	4337995.51
164	48589009	Minor depressive disorder (disorder)	601373.48
100	370143000	Major depressive disorder	343371.34
99	66214007	Substance abuse (disorder)	42708.6
34	91302008	Sepsis (disorder)	14667.6
30	389087006	Hypoxemia (disorder)	12942
27	414545008	Ischemic heart disease (disorder)	43626.6
25	274531002	Abnormal findings diagnostic imaging heart+coronary circulat (finding)	87833.79

A Clinician - A

mysql

31 0 affected 0.604 s

Download

6. Researcher1 accesses HospitalA's database

[illegible]

7. Researcher1 selects from patients table at HospitalA

Researcher1 - A - Beekeeper Studio

select patients

select procedures (mysql)

1 select * from patients;

Save

Run

birthdate	deathdate	ssn	drivers	passport	prefix	first	last	suffix	maiden
1993-01-12	2022-07-24	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
1976-09-10	(EMPTY)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
1956-12-14	(EMPTY)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
1961-07-05	(EMPTY)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
1931-08-24	2015-10-06	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
1975-04-23	(EMPTY)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
1939-03-18	(EMPTY)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
1959-01-10	(EMPTY)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
1957-01-12	(EMPTY)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
1955-04-20	(EMPTY)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

Researcher1 - A

mysql

118 0 affected 0.119 s

Download

7. Researcher1 selects from patients table at HospitalA (cont)

A

Filter

ENTITIES 9

conditions

encounters

medications

observations

organizations

patients

payers

procedures

providers

Researcher1 - A - Beekeeper Studio

select patients

select procedures (mysql)

1 select * from patients;

Save

Run

county	fips	zip	lat	lon	healthcare_expenses	healthcare_coverage	income
LL	(NULL)	(NULL)	(NULL)	(NULL)	56459.65	536860.07	65649
LL	(NULL)	(NULL)	(NULL)	(NULL)	41250.84	515102.22	149834
LL	(NULL)	(NULL)	(NULL)	(NULL)	49734.31	222131.89	105372
LL	(NULL)	(NULL)	(NULL)	(NULL)	80570.12	1909435.57	71541
LL	(NULL)	(NULL)	(NULL)	(NULL)	53072.10	96125.16	115790
LL	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
LL	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
LL	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
LL	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

Researcher1 - A

mysql

118 0 affected 0.75 s

Download

8. Researcher1 selects from procedures table at HospitalA

A

Filter

ENTITIES 9

conditions

encounters

medications

observations

organizations

patients

payers

procedures

providers

select patients

select procedures (mysql)

```
1 select
2   count(*) as count,
3   reasoncode,
4   reasondescription,
5   round(sum(procedures.base_cost), 2)
6 from procedures
7 where patient in (select patient from encounters where reasoncode = '91302008')
8 group by reasoncode, reasondescription
9 order by count desc;
```

Save

Run

count	reasoncode	reasondescription	round(sum(procedures.bas
1714	(EMPTY)	(EMPTY)	2286703.11
99	66214007	Substance abuse (disorder)	42708.6
34	91302008	Sepsis (disorder)	14667.6
30	389087006	Hypoxemia (disorder)	12942
27	414545008	Ischemic heart disease (disorder)	1855.02
25	274531002	Abnormal findings diagnostic imaging heart+coronary circulat (finding)	44800.49
16	88805009	Chronic congestive heart failure (disorder)	(NULL)
14	10509002	Acute bronchitis (disorder)	2081.88

Researcher1 - A

mysql

29 0 affected 1.94 s

Download

9. B_clinician selects from treatments table at HospitalB

B

Filter

ENTITIES 197

public

diagnoses

insurance_plans

organizations

patients

prescriptions

providers

reports

treatments

visits

information_schema

pg_catalog

select treatments (postgres)

insert treatment (postgres)

select new treatment (postgres)

```
1 select
2   count(treatments.id) as count,
3   treatments.reasoncode,
4   treatments.reasondescription,
5   round(sum(treatments.base_cost), 2)
6 from treatments
7 where patient in (select patient from visits where reasoncode = '91302008')
8 group by reasoncode, reasondescription
9 order by count desc;
```

Save

Run

count	reasoncode	reasondescription	round
952	(NULL)	(NULL)	1445962.88
169	48589009	Minor depressive disorder (disorder)	573406.12
39	389087006	Hypoxemia (disorder)	16824.60
22	66214007	Substance abuse (disorder)	2708.64
21	185086009	Chronic obstructive bronchitis (disorder)	131765.90
10	10509002	Acute bronchitis (disorder)	30591.11
9	39898005	Sleep disorder (disorder)	3882.60
8	91302008	Sepsis (disorder)	3451.20
6	(EMPTY)	(EMPTY)	0.00

B Clinician - B

postgresql

21 0 affected 2.768 s

Download

10. **Researcher1** accesses **HospitalB's** database

[illegible]

11. Researcher1 selects from treatments table at HospitalB

B

Filter

ENTITIES 197

public

diagnoses

insurance_plans

organizations

patients

prescriptions

providers

reports

treatments

visits

information_schema

pg_catalog

select patients

select treatments (postgres)

select new treatment (postgres)

```
1 select
2   count(treatments.id) as count,
3   treatments.reasoncode,
4   treatments.reasondescription,
5   round(sum(treatments.base_cost), 2)
6 from treatments
7 where patient in (select patient from visits where reasoncode = '91302008')
8 group by reasoncode, reasondescription
9 order by count desc;
```

Save

Run

count	reasoncode	reasondescription	round
145	(NULL)	(NULL)	898550.78
9	39898005	Sleep disorder (disorder)	3882.60
8	91302008	Sepsis (disorder)	3451.20
4	10509002	Acute bronchitis (disorder)	15014.31
2	65966004	Fracture of forearm	8894.91
2	67782005	Acute respiratory distress syndrome (disorder)	862.80
1	444470001	Injury of anterior cruciate ligament	9139.34
1	283371005	Laceration of forearm	9990.32
1	195662009	Acute viral pharyngitis (disorder)	2161.14

Researcher1 - B

postgres

9 0 affected 0.998 s

Download

12. B_clinician inserts new mental health treatment row

B

▼ ↺ +

Filter ▼

ENTITIES 197 +

▼ public

> diagnoses

> insurance_plans

> organizations

> patients

> prescriptions

> providers

> reports

> treatments

> visits

> information_schema

> pg_catalog

select treatments (postgres)

insert treatment (postgres)

select new treatment (postgres)

+

```
1 insert into treatments (id, patient, visit, start, stop, code, description,
2   base_cost, reasoncode, reasondescription)
3 values(
4   12762, 'af072a13-3c11-91fc-6d87-6fbad23d1981',
5   'efb57289-2357-c1a0-b1e9-838c13db9ac7', NOW(), NOW(), '171207006',
6   'Depression screening (procedure)', 123, '48589009',
7   'Minor depressive disorder (disorder)'
8 );
```

Save Run ▼

Query 1/1: No Results. 1 rows affected. See the select box in the bottom left for more query results.

B Clinician - B

postgresql

1 affected 0.708 s

Download ▼

12. B_clinician inserts new mental health treatment record (cont)

B

Filter

ENTITIES 197

public

diagnoses

insurance_plans

organizations

patients

prescriptions

providers

reports

treatments

visits

information_schema

pg_catalog

select treatments (postgres)

insert treatment (postgres)

select new treatment (postgres)

1 select * from treatments where id=12762

Save Run

	code	description	base_cost	reasoncode	reasondescription
4	171207006	Depression screening (procedure)	123	48589009	Minor depressive disorder (disorder)

B Clinician - B

postgresql

1 0 affected 0.465 s

Download

13. Researcher1 selects new row

B

▼ ↺ +

Filter

▼

ENTITIES 197

⌵ ↺ +

▼ public

> diagnoses ⭐

> insurance_plans

> organizations

> patients

> prescriptions

> providers

> reports

> treatments

> visits

> information_schema

> pg_catalog

<>

1 select * from treatments where id = 12762;

Save

Run ▼

Query 1/1: No Results. 0 rows affected. See the select box in the bottom left for more query results.

Researcher1 - B

postgresql

0 affected

0.061 s

Download ▼

14. HospitalB admin revokes Researcher1's access

Research Institute Access Dashboard

?

Requests

⋮

Policies

🔍

Search

👤

researcher1

Policies

Requester

researcher1

Target Org

HospitalA

15. Researcher1 selects from treatments table at HospitalB

B

▼ ↺ +

Filter

▼

ENTITIES 197

+

▼ public

> diagnoses

> insurance_plans

> organizations

> patients

> prescriptions

> providers

> reports

> treatments

> visits

> information_schema

> pg_catalog

select patients

select treatments (postgres

select new treatment (postg

+

```
1 select
2   count(treatments.id) as count,
3   treatments.reasoncode,
4   treatments.reasondescription,
5   round(sum(treatments.base_cost), 2)
6 from treatments
7 where patient in (select patient from visits where reasoncode = '91302008')
8 group by reasoncode, reasondescription
9 order by count desc;
```

Save

Run

▼

Query 1/1: No Results. 0 rows affected. See the select box in the bottom left ✓
for more query results.

Researcher1 - B

postgresql

0 affected 0.181 s

Download ▼

Conclusion and Summary

- The ability to share database resources among collaborating organizations is highly desirable.
- However, challenges persist regarding interoperability in the exchange of resources between organizations and the preservation of local access policies.
- SFDS provides a generic data sharing infrastructure that effectively and securely achieves data sharing objectives.
- Transparent to the otherwise normal business operations of participating organizations.
- This ease of deployment, granularity of control and efficiency make this new infrastructure solution practical for meeting the data sharing and protection objectives of the clinical research community.

Future Work

- The current focus of SFDS is to allow sharing of database resources. Included in our plans is to extend the infrastructure to allow controlled access to non-structured data such as files.
- We would like to organize a full-scale pilot study involving institutions that house medical information and/or conduct clinical research.
- Our goal is to transition SFDS from a research project to operational use.



Thank You

Joshua Roberts and David Ferraiolo

*Acknowledgement: D. Chris Compton
and Joanna DeFranco*



Analytics with Altitude
2023 HDAA Conference
Denver, Colorado



Children's Hospital Colorado

