MedChain

Blockchain-based Access Control System for Medical Data

Muhammad Talha Paracha
supervised by
Dr. Troncoso-Pastoriza & **Dr. Raisaro**
with
Prof. Hubaux

About me

Recently graduated from NUST, Pakistan
as a Software Engineer

About me

Recently graduated from NUST, Pakistan as a Software Engineer

Working at LCA1 for the past 2.5 months

About me

Recently graduated from NUST, Pakistan as a Software Engineer

Working at LCA1 for the past 2.5 months

Aspire to become a computer scientist

About the talk

Medical data-analysis

About the talk

Medical data-analysis

Blockchains

About the talk

Medical data-analysis

Blockchains

MedChain

- Design
- Integration
- Demo
- Performance evaluation

About the talk

Medical data-analysis

Blockchains

MedChain
- Design
- Integration
- Demo
- Performance evaluation

Conclusion

Medical data-analysis

Hospital 1

xml

Hospital 2

i2b2

...

Hospital N

ElasticSearch

Hospital 1

xml

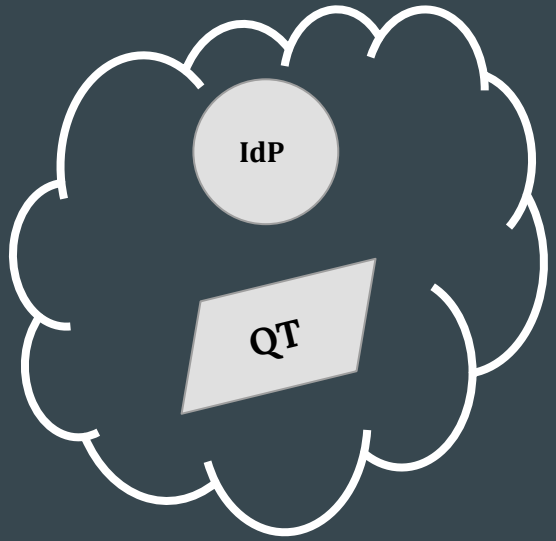
Hospital 2

i2b2

...

Hospital N

ElasticSearch



Hospital 1

xml

Hospital 2

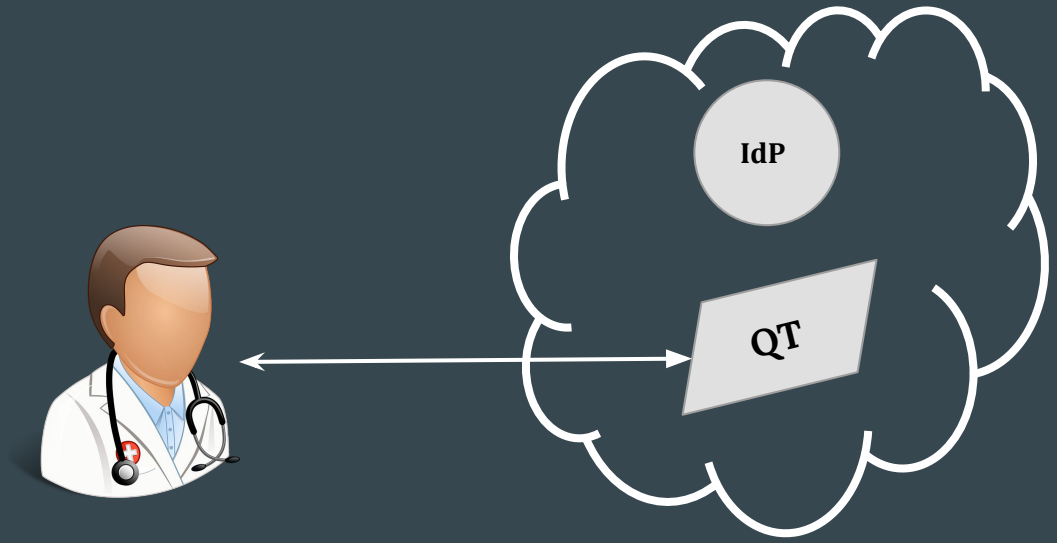
i2b2

...

Hospital N

ElasticSearch

IdP = Identity Provider
QT = Query Tool



Hospital 1

xml

Hospital 2

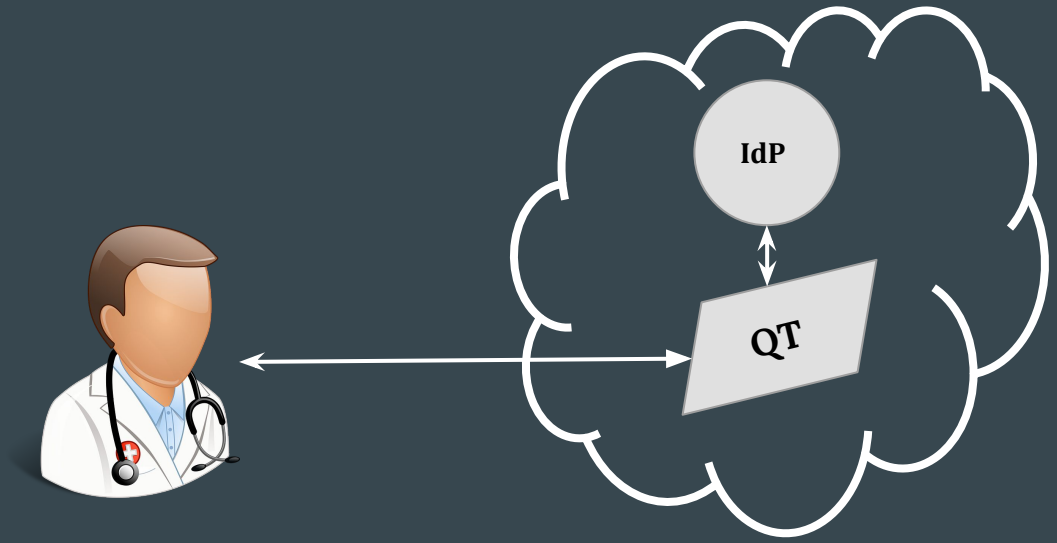
i2b2

...

Hospital N

ElasticSearch

IdP = Identity Provider
QT = Query Tool



Hospital 1

xml

Hospital 2

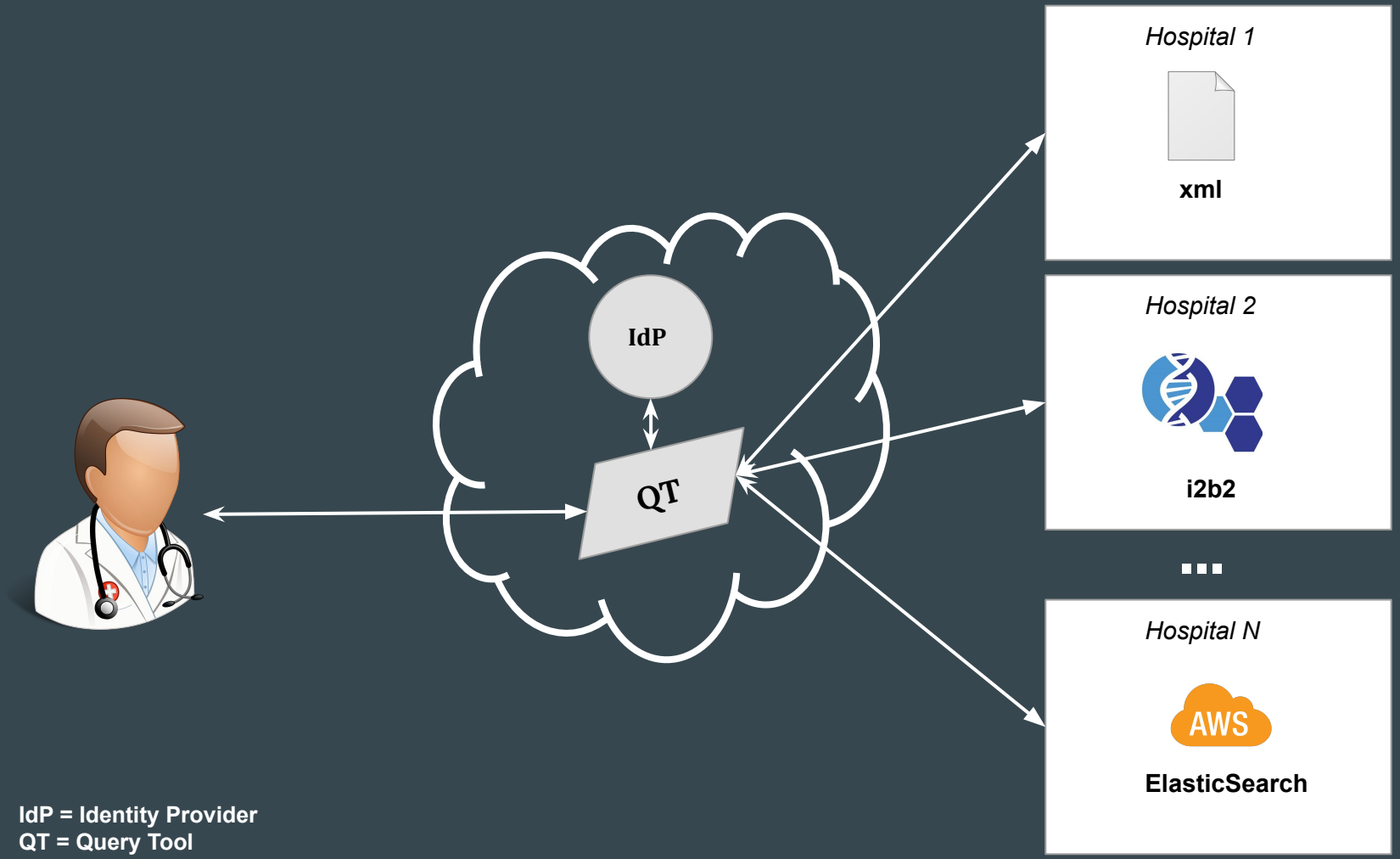
i2b2

...

Hospital N

ElasticSearch

IdP = Identity Provider
QT = Query Tool



IdP = Identity Provider
QT = Query Tool

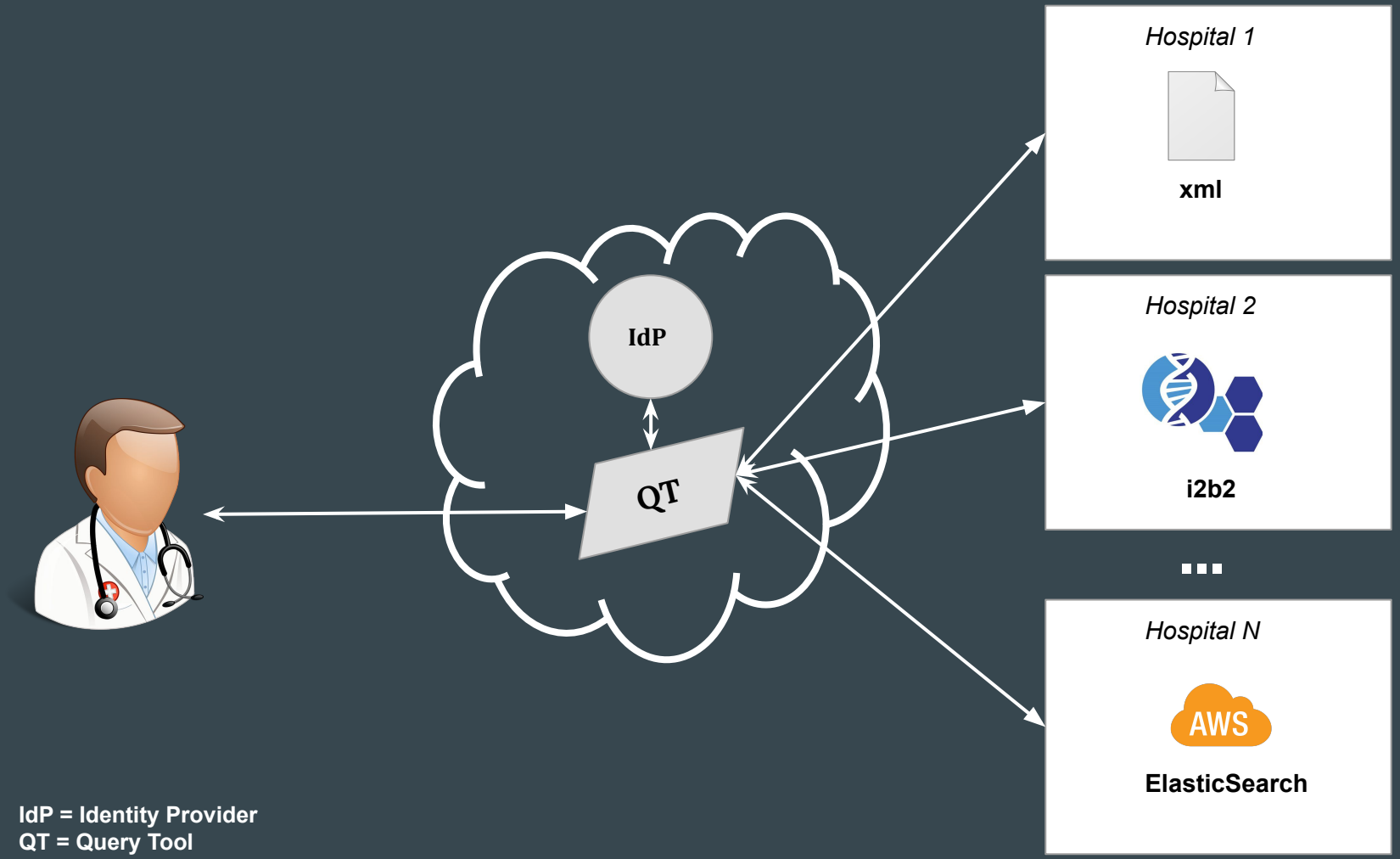
Threat model

- Malicious external agents
- Dishonest internal agents

IdP = Identity Provider

Vulnerabilities of current approaches

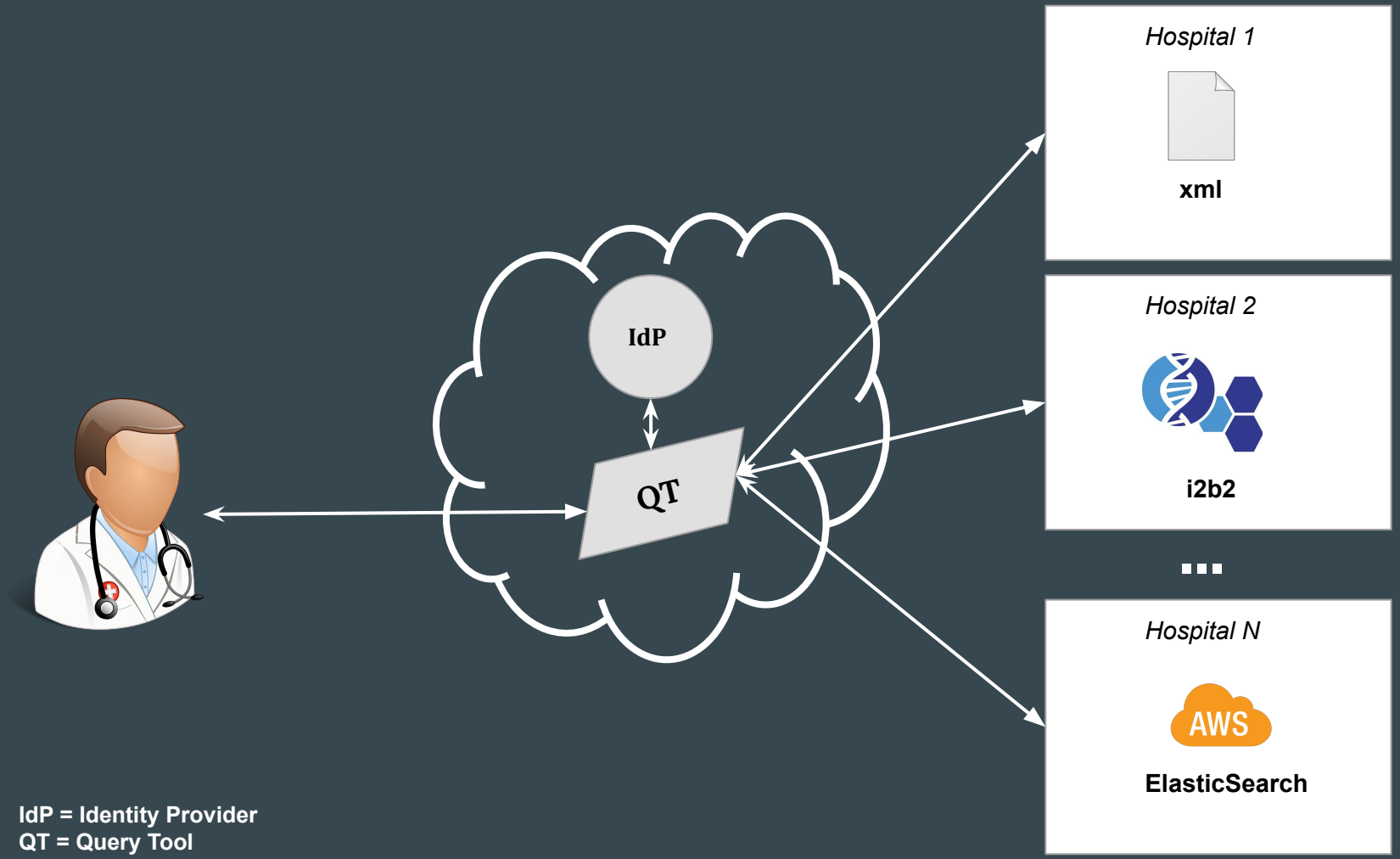
Single Point of Failure	• IdP can be compromised



IdP = Identity Provider
QT = Query Tool

Vulnerabilities of current approaches

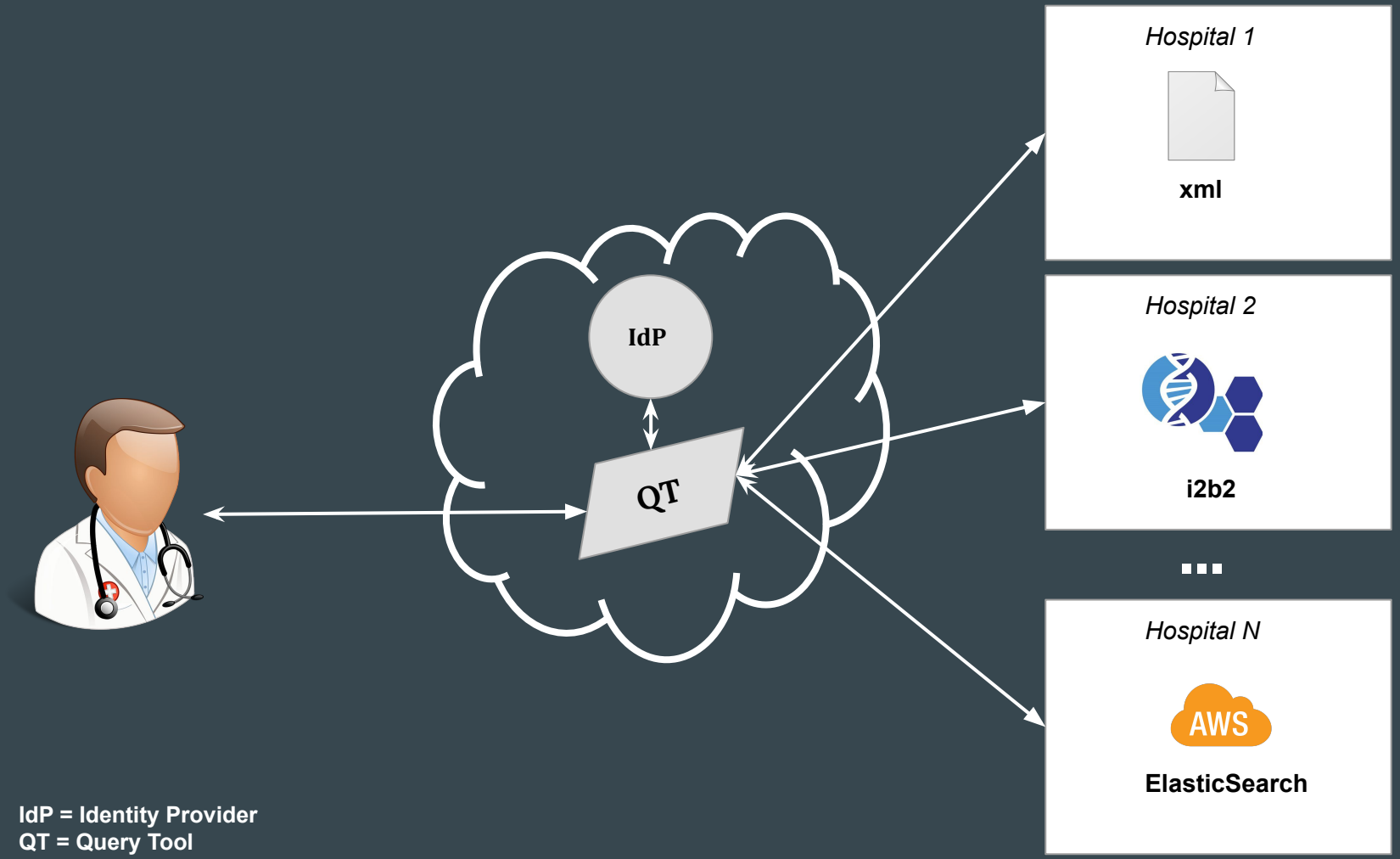
Single Point of Failure	• IdP can be compromised
No Auditability (by the resources)	• Resources are not aware of the end-user who wants to access the data.



IdP = Identity Provider
QT = Query Tool

Vulnerabilities of current approaches

Single Point of Failure	• IdP can be compromised
No Auditability (by the resources)	• Resources are not aware of the end-user who wants to access the data
No Authorization (only Authentication)	• Access control policies for different users cannot be specified



IdP = Identity Provider
QT = Query Tool

Project objective:

Build a **decentralized**, **auditable, immutable** access control system for medical data analysis

Blockchains

DEDIS Omniledger Implementation

- Inspired by the research on
  - **ByzCoin**: Byzantine fault-tolerant collectively signed transactions.

DEDIS Omniledger Implementation

- Inspired by the research on
  - **ByzCoin**: Byzantine fault-tolerant collectively signed transactions.
  - **CHAINIAC**: Forward-links for fast verification of data-on-the-chain.

DEDIS Omniledger Implementation

- Inspired by the research on
  - **ByzCoin**: Byzantine fault-tolerant collectively signed transactions.
  - **CHAINIAC**: Forward-links for fast verification of data-on-the-chain.
  - **Omniledger**: Sharding for scale-out transactions.*

DEDIS Omniledger Implementation

- Inspired by the research on
  - **ByzCoin**: Byzantine fault-tolerant collectively signed transactions.
  - **CHAINIAC**: Forward-links for fast verification of data-on-the-chain.
  - **Omniledger**: Sharding for scale-out transactions.*
- Uses smart-contracts precompiled in code.

DEDIS Omniledger Implementation

- Inspired by the research on
  - **ByzCoin**: Byzantine fault-tolerant collectively signed transactions.
  - **CHAINIAC**: Forward-links for fast verification of data-on-the-chain.
  - **Omniledger**: Sharding for scale-out transactions.*
- Uses smart-contracts precompiled in code.
- Permissioned-blockchain.

DEDIS Omniledger Implementation

- Inspired by the research on
  - **ByzCoin**: Byzantine fault-tolerant collectively signed transactions.
  - **CHAINIAC**: Forward-links for fast verification of data-on-the-chain.
  - **Omniledger**: Sharding for scale-out transactions.*
- Uses smart-contracts precompiled in code.
- Permissioned-blockchain.
- Introduces **Distributed Access Rights Control (DARC)** data-structure.

DARCs

- Mapping of actions to expressions.

DARCs

- **Mapping of actions to expressions.**
  - doSomething: identityA

DARCs

- Mapping of actions to expressions.
  - doSomething: identityA
- **Rules can be evolved**

DARCs

- Mapping of actions to expressions.
  - doSomething: identityA
- **Rules can be evolved**
  - evolve: systemAdmin

DARCs

- Mapping of actions to expressions.
  - doSomething: identityA
- **Rules can be evolved**
  - evolve: systemAdmin
  - doSomething: identityA → doSomething: identityB

DARCs

- Mapping of actions to expressions.
  - doSomething: identityA
- Rules can be evolved
  - evolve: systemAdmin
  - doSomething: identityA → doSomething: identityB
- **Rules can contain multiple identities**

DARCs

- Mapping of actions to expressions.
  - doSomething: identityA
- Rules can be evolved
  - evolve: systemAdmin
  - doSomething: identityA → doSomething: identityB
- **Rules can contain multiple identities**
  - doSomething: identityA **AND** (identityB **OR** identityC)

DARCs

- Mapping of actions to expressions.
  - doSomething: identityA
- Rules can be evolved
  - evolve: systemAdmin
  - doSomething: identityA → doSomething: identityB
- Rules can contain multiple identities
  - doSomething: identityA **AND** (identityB **OR** identityC)
- **Rules can reference other DARCs**

DARCs

- Mapping of actions to expressions.
  - doSomething: identityA
- Rules can be evolved
  - evolve: systemAdmin
  - doSomething: identityA → doSomething: identityB
- Rules can contain multiple identities
  - doSomething: identityA **AND** (identityB **OR** identityC)
- **Rules can reference other DARCs**

DARCs

- Mapping of actions to expressions.
  - doSomething: identityA
- Rules can be evolved
  - evolve: systemAdmin
  - doSomething: identityA → doSomething: identityB
- Rules can contain multiple identities
  - doSomething: identityA **AND** (identityB **OR** identityC)
- **Rules can reference other DARCs**
  - doSomethingGroup1: identityA
  - doSomethingGroup2: identity B **OR** C
  - doSomething: doSomethingGroup1 **AND** doSomethingGroup2

MedChain:

Omniledger with predefined contracts (and DARCs) suitable for providing an access control system for medical data analysis

MedChain: Design

Hospital 1

Hospital 2

Hospital 1

Hospital 2

Admin1

Admin2

Hospital 1

Hospital 2

Admin1

Manager1x

Manager1y

Manager2x

Manager2y

Admin2

Hospital 1

Hospital 2

Admin1

Manager1x

Manager1y

User1x

User1y

Manager2x

Manager2y

Admin2

User2x

User2y

Admins: A1, A2
Managers: M1x, M1y, M2x, M2y
Users: U1x, U1y, U2x, U2x

Admins: A1, A2
Managers: M1x, M1y, M2x, M2y
Users: U1x, U1y, U2x, U2x

Genesis

evolve: A1 & A2

create_darc: A1 & A2

Admins: A1, A2
Managers: M1x, M1y, M2x, M2y
Users: U1x, U1y, U2x, U2x

Genesis

evolve: A1 & A2

create_darc: A1 & A2

Managers1

evolve: A1

include: M1x & M1y

Admins: A1, A2
Managers: M1x, M1y, M2x, M2y
Users: U1x, U1y, U2x, U2x

Genesis

evolve: A1 & A2

create_darc: A1 & A2

Managers1

evolve: A1

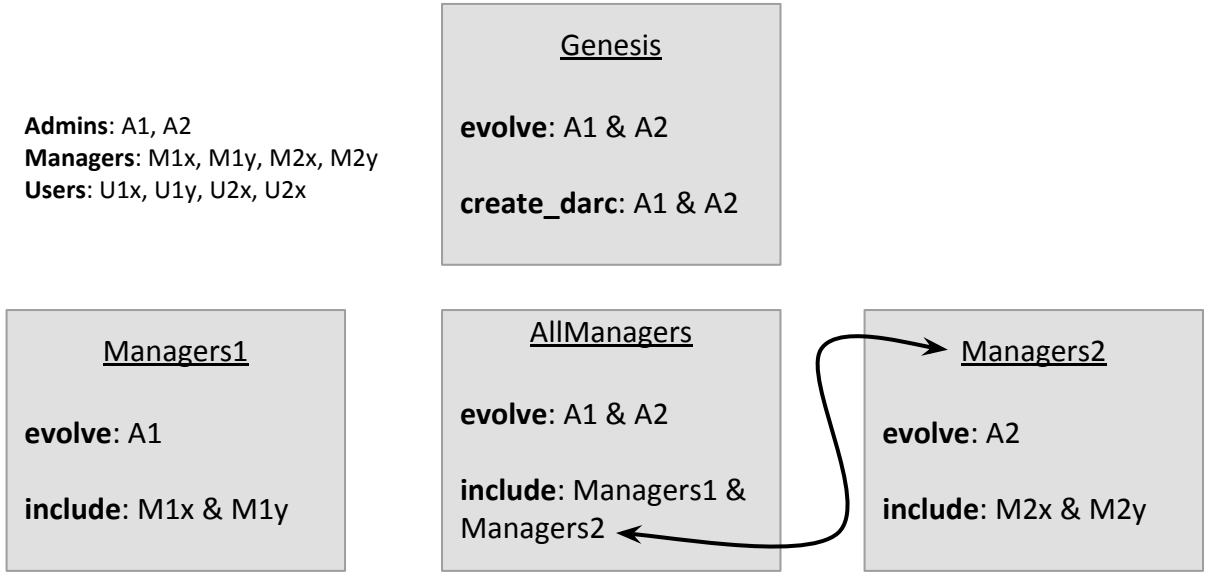
include: M1x & M1y

Managers2

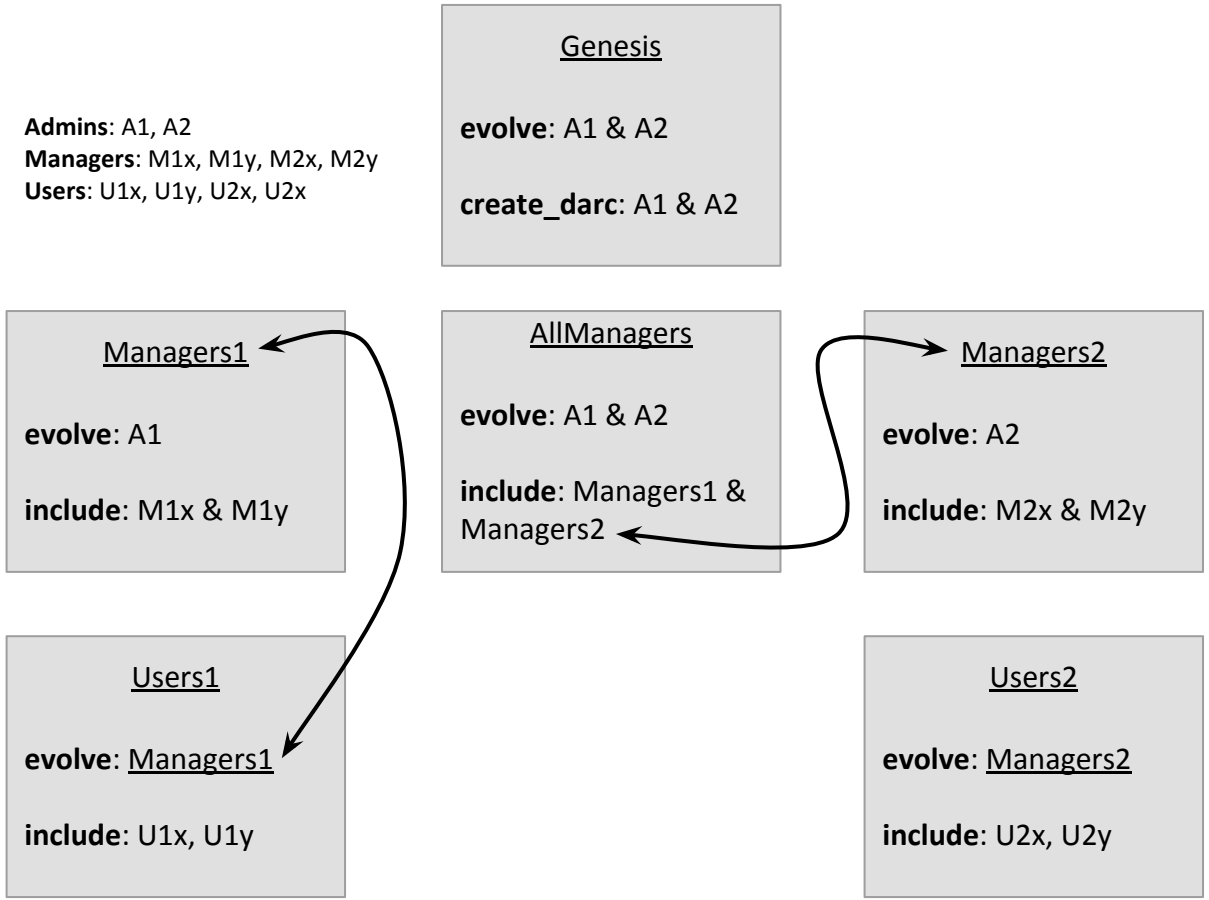
evolve: A2

include: M2x & M2y

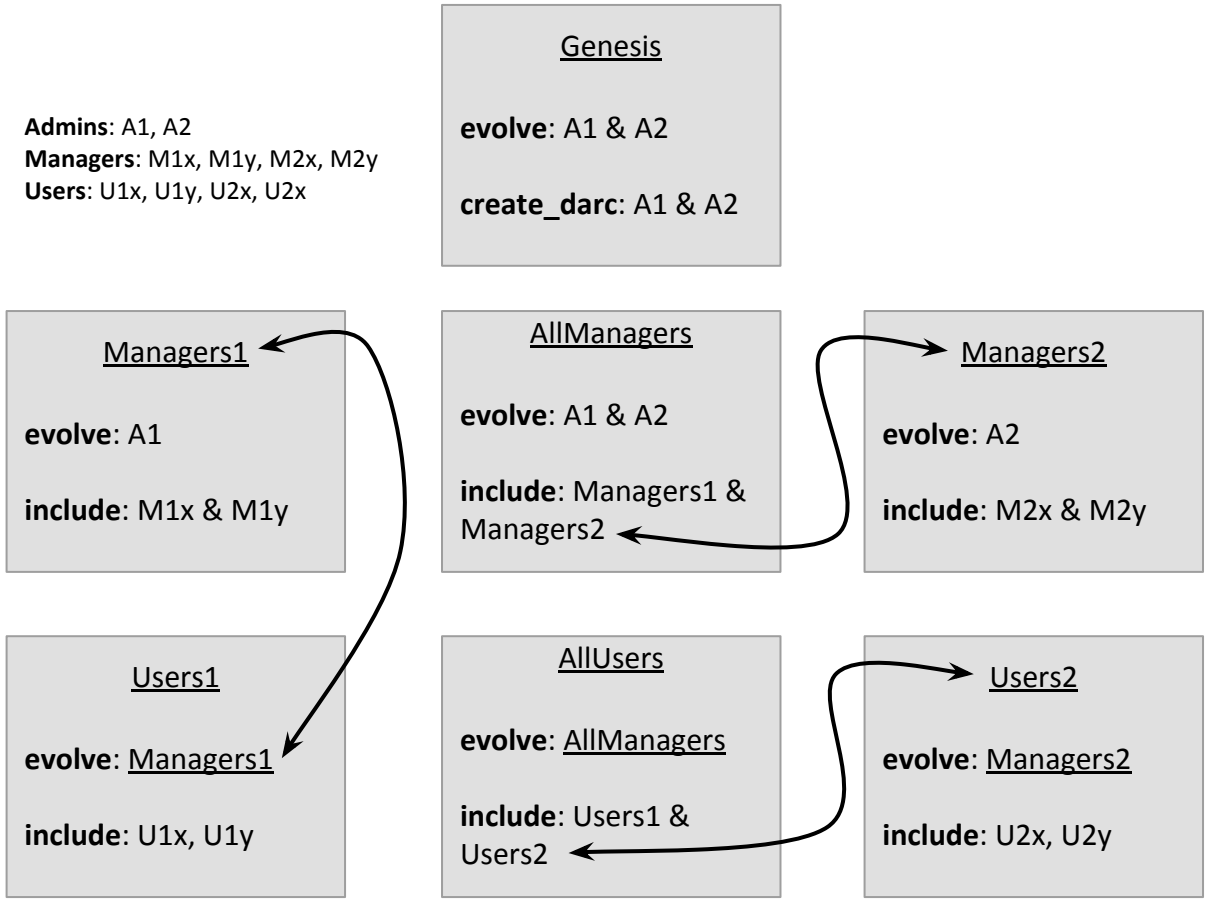
Admins: A1, A2
Managers: M1x, M1y, M2x, M2y
Users: U1x, U1y, U2x, U2x

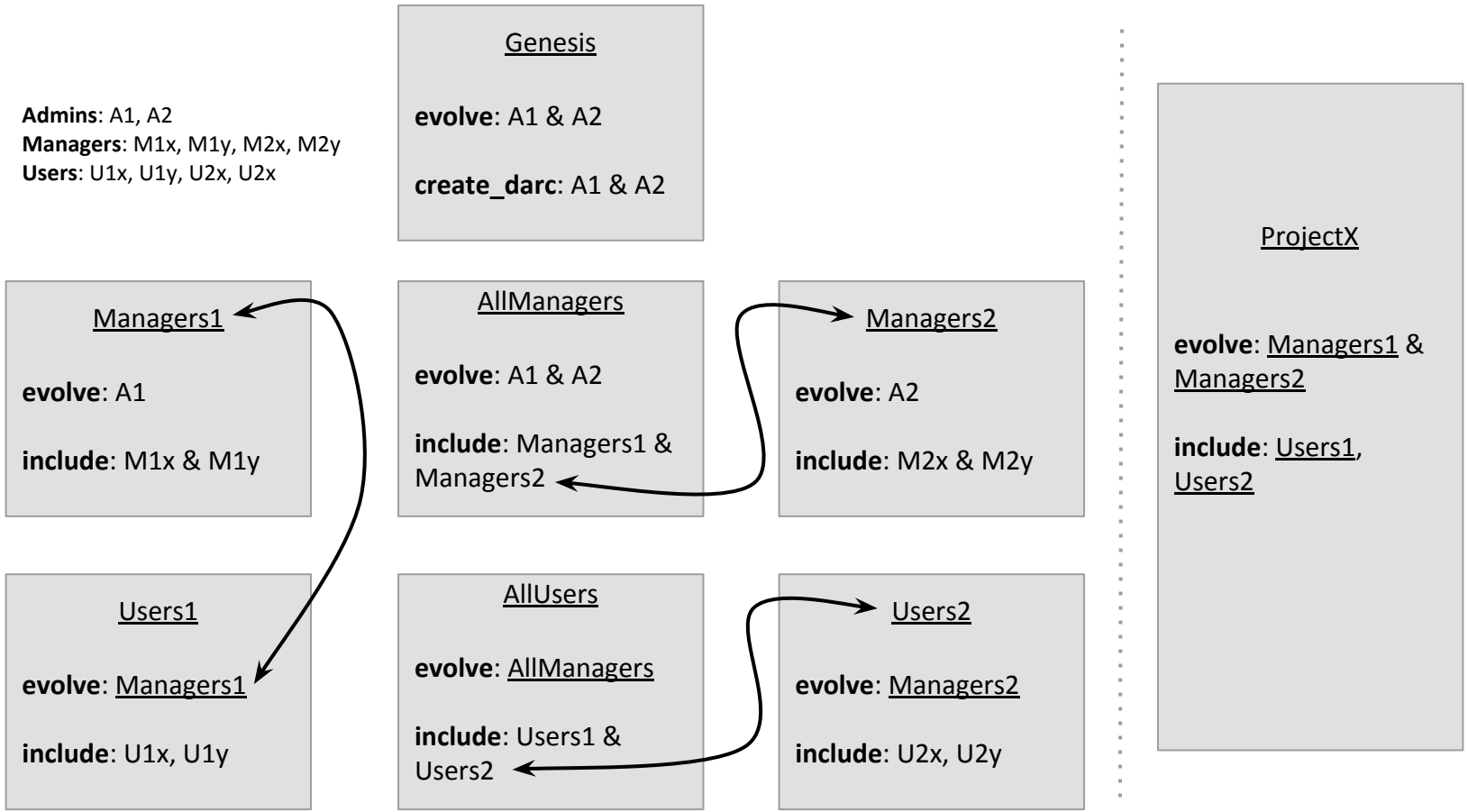


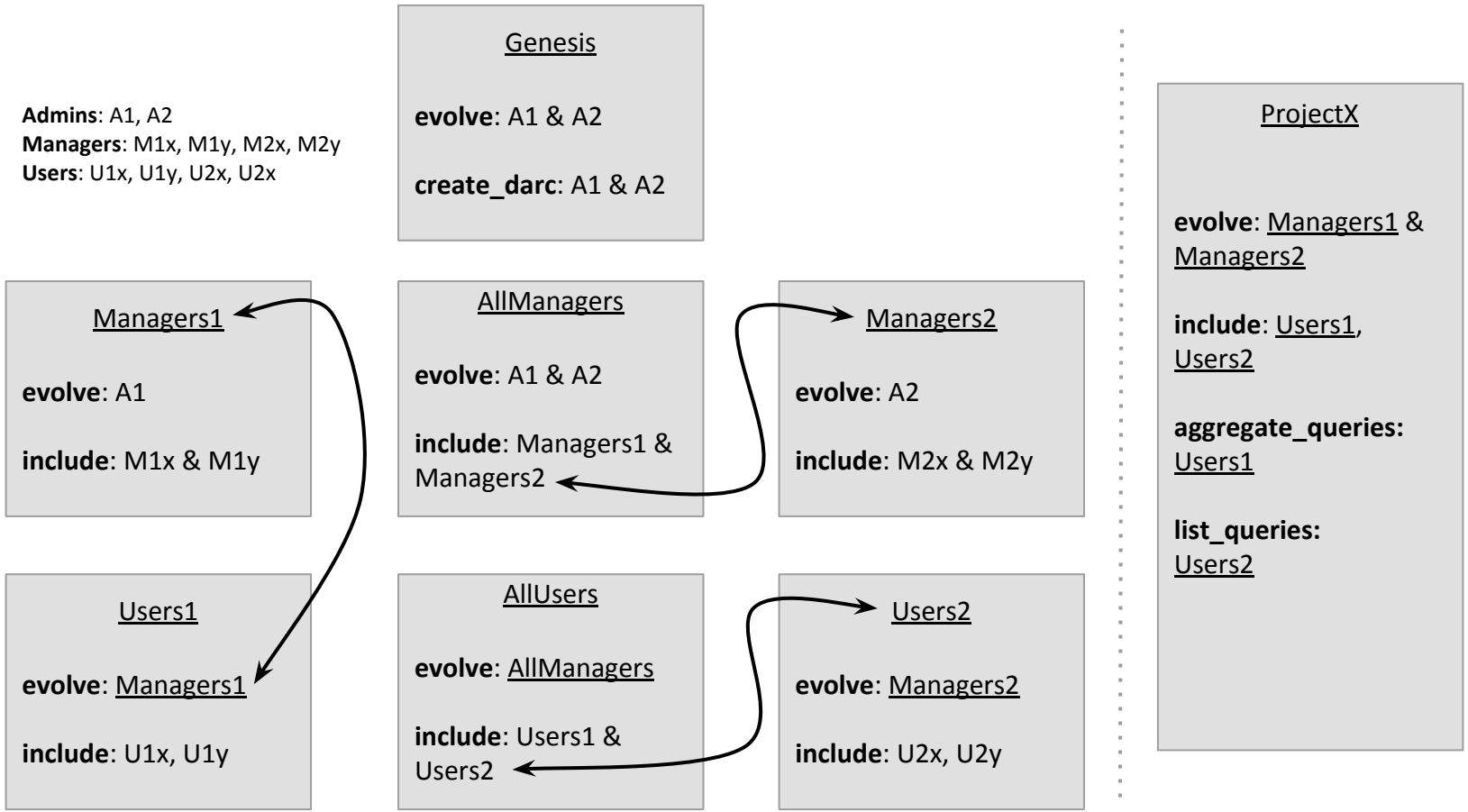
Admins: A1, A2
Managers: M1x, M1y, M2x, M2y
Users: U1x, U1y, U2x, U2x



Admins: A1, A2
Managers: M1x, M1y, M2x, M2y
Users: U1x, U1y, U2x, U2x




Admins: A1, A2
Managers: M1x, M1y, M2x, M2y
Users: U1x, U1y, U2x, U2x
Genesis
evolve: A1 & A2
create_darc: A1 & A2
Managers1
evolve: A1
include: M1x & M1y
AllManagers
evolve: A1 & A2
include: Managers1 & Managers2
Managers2
evolve: A2
include: M2x & M2y
Users1
evolve: Managers1
include: U1x, U1y
AllUsers
evolve: AllManagers
include: Users1 & Users2
Users2
evolve: Managers2
include: U2x, U2y
ProjectX
evolve: Managers1 & Managers2
include: Users1, Users2


Admins: A1, A2
Managers: M1x, M1y, M2x, M2y
Users: U1x, U1y, U2x, U2x
Genesis
evolve: A1 & A2
create_darc: A1 & A2
Managers1
evolve: A1
include: M1x & M1y
AllManagers
evolve: A1 & A2
include: Managers1 & Managers2
Managers2
evolve: A2
include: M2x & M2y
Users1
evolve: Managers1
include: U1x, U1y
AllUsers
evolve: AllManagers
include: Users1 & Users2
Users2
evolve: Managers2
include: U2x, U2y
ProjectX
evolve: Managers1 & Managers2
include: Users1, Users2
aggregate_queries: Users1
list_queries: Users2

MedChain: Design (contd.)

- We develop contracts for validating access control;
  - getListOfProjects
  - runQuery

MedChain: Design (contd.)

- We develop contracts for validating access control;
  - getListOfProjects
  - runQuery

Contracts, after successfully validating access control, append information to the blockchain (which provides an immutable log of the system events).

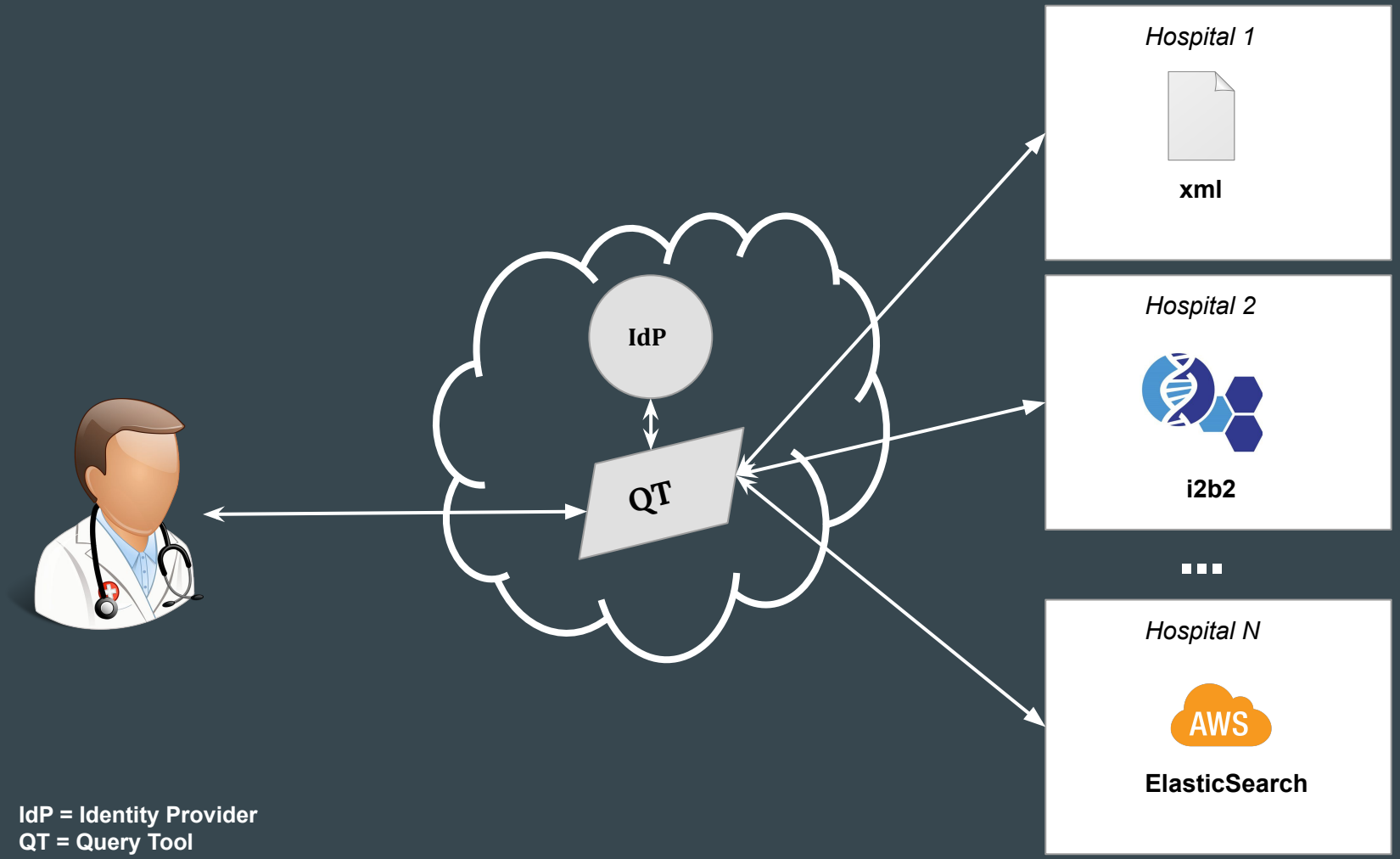
MedChain: Design (contd.)

- We develop contracts for validating access control;
  - getListOfProjects
  - runQuery

Contracts, after successfully validating access control, append information to the blockchain (which provides an immutable log of the system events).

Clients can then interact with the blockchain to retrieve the information logged (along with a proof).

MedChain: Integration



IdP = Identity Provider
QT = Query Tool

Inter-Resource Communication Tool (IRCT)

- Developed at the Harvard University Medical School

https://pic-sure.org/products/bd2k-pic-sure-restful-api

Inter-Resource Communication Tool (IRCT)

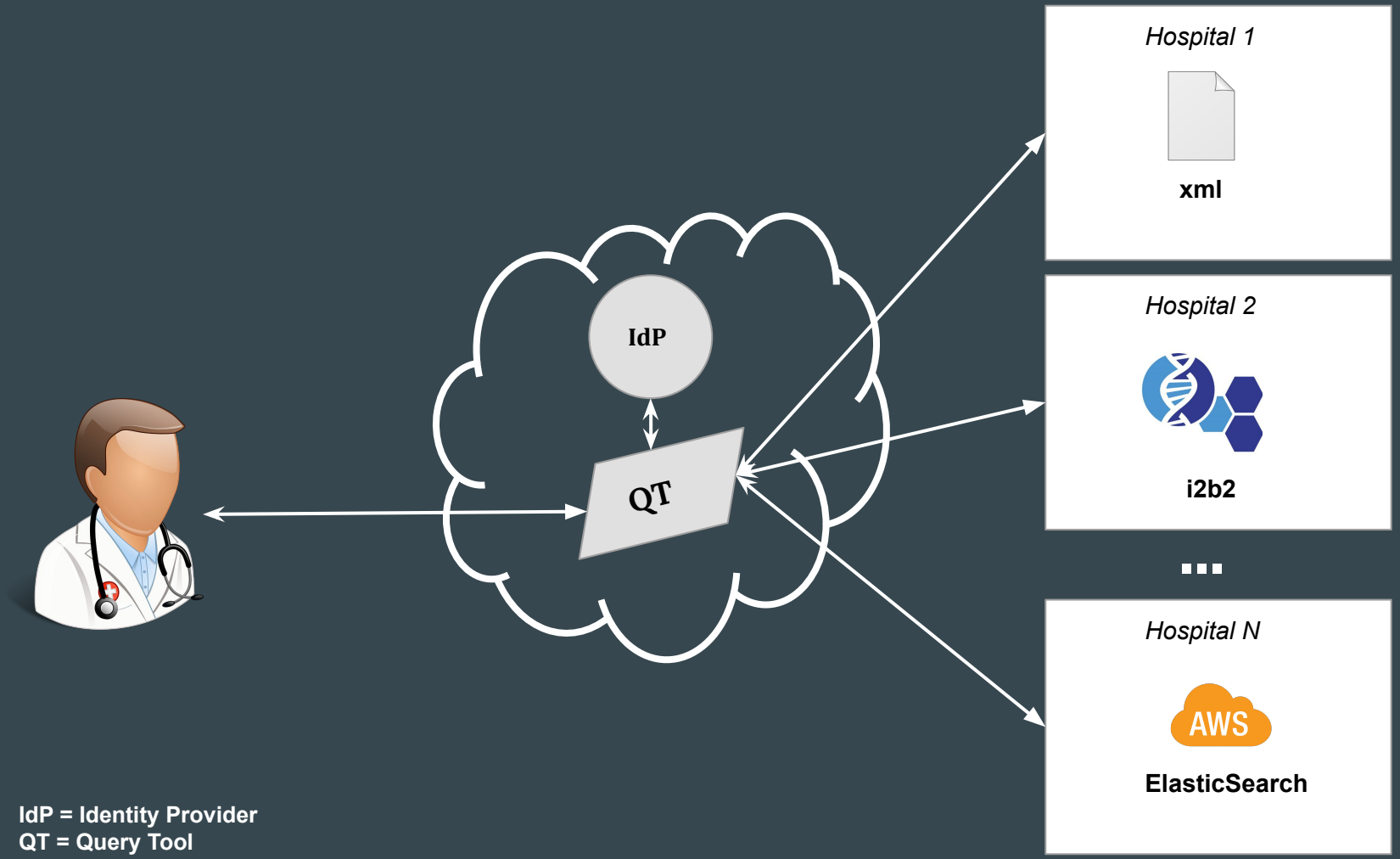
- Developed at the Harvard University Medical School
- **Open-source infrastructure for biomedical research**

https://pic-sure.org/products/bd2k-pic-sure-restful-api

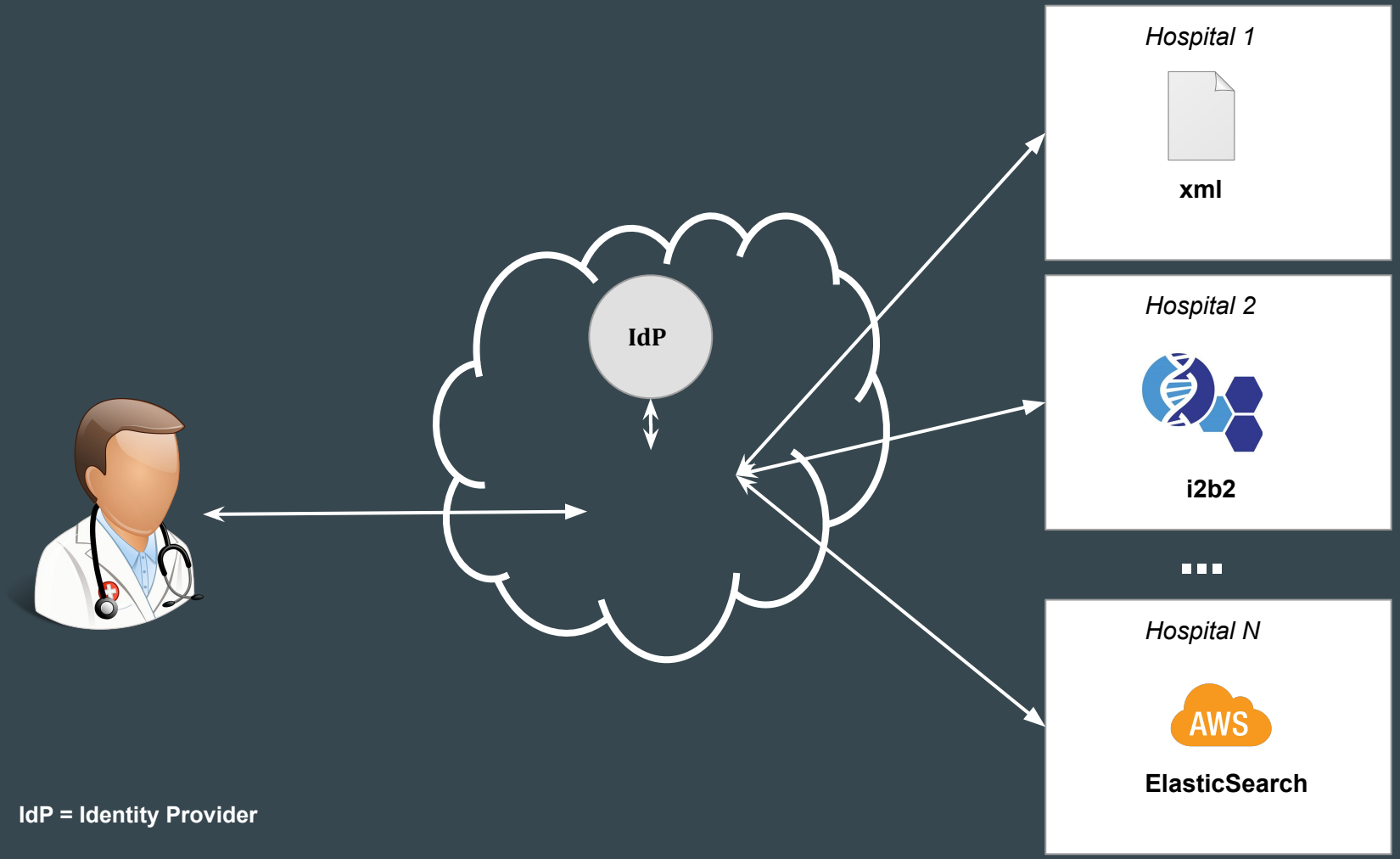
Inter-Resource Communication Tool (IRCT)

- Developed at the Harvard University Medical School
- Open-source infrastructure for biomedical research
- **Supports multiple heterogenous patient-level datasets**

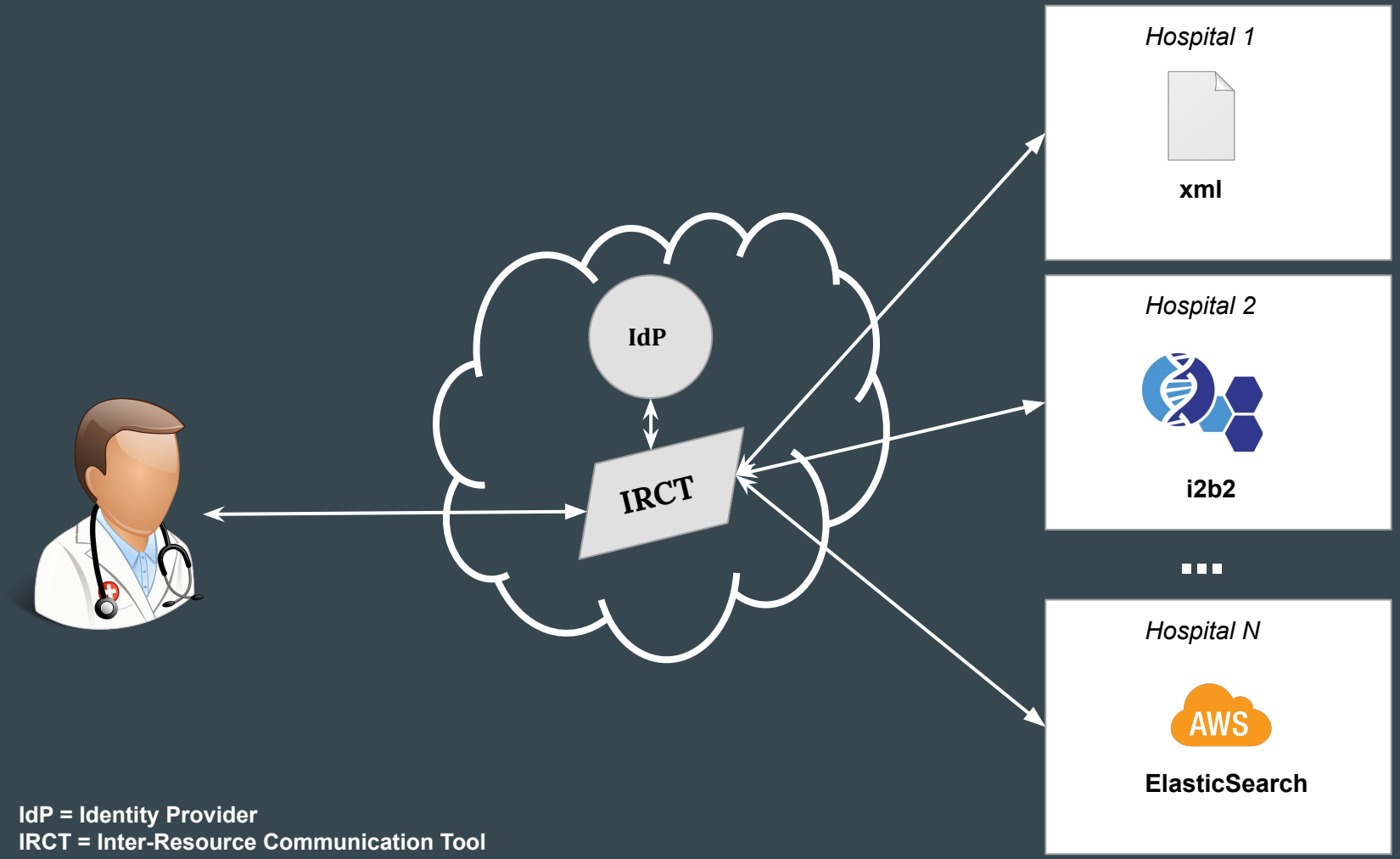
https://pic-sure.org/products/bd2k-pic-sure-restful-api



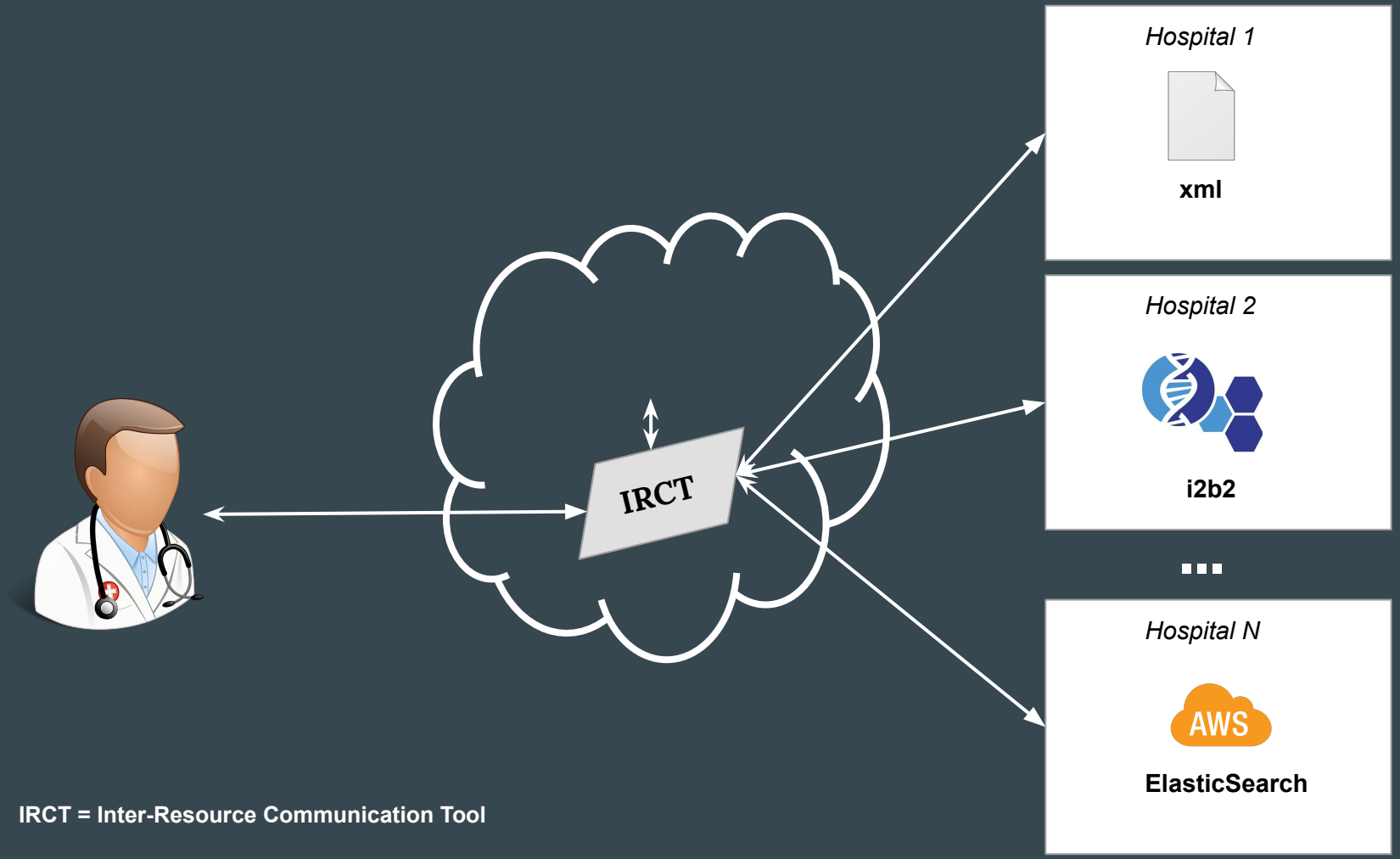
IdP = Identity Provider
QT = Query Tool



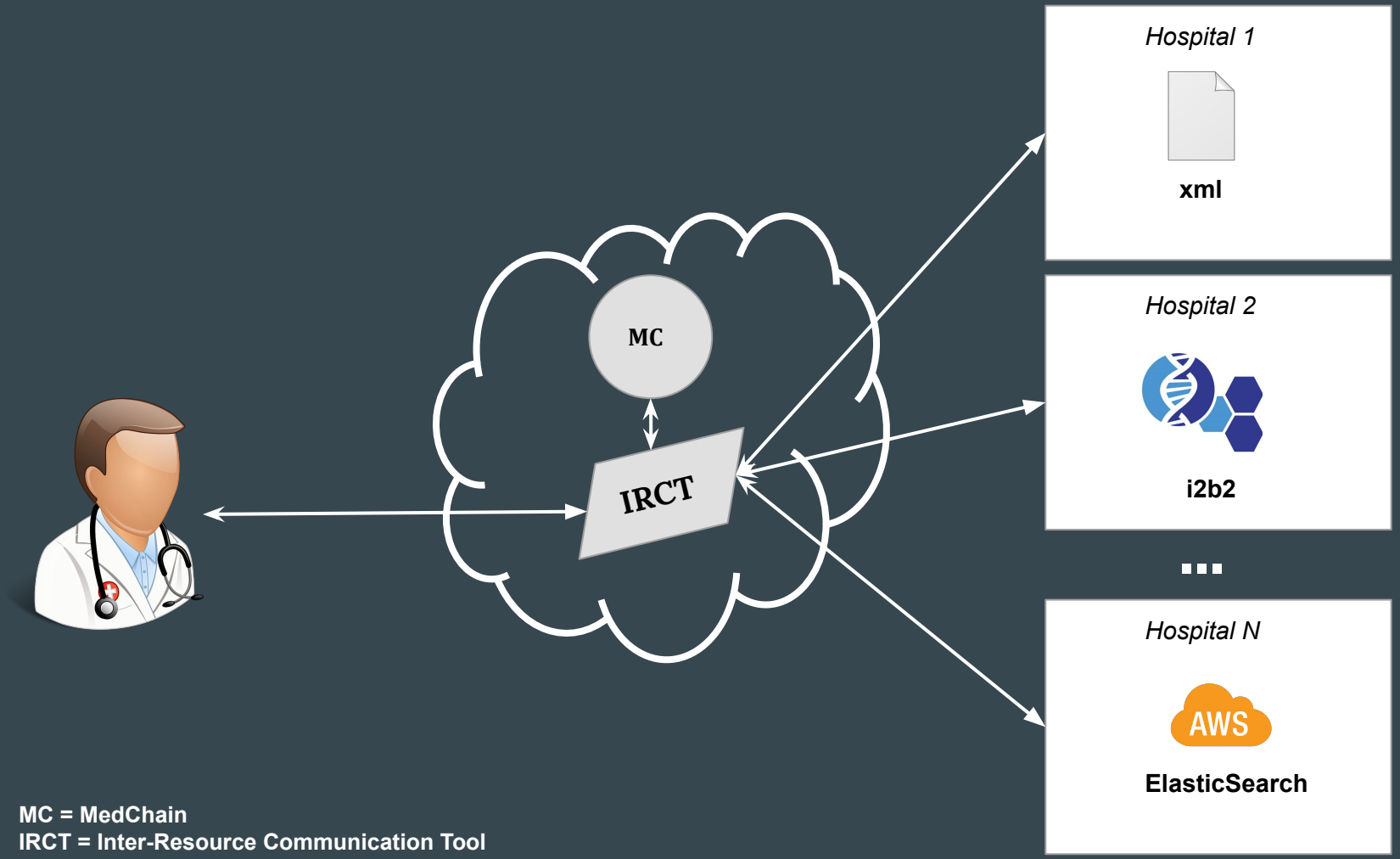
IdP = Identity Provider



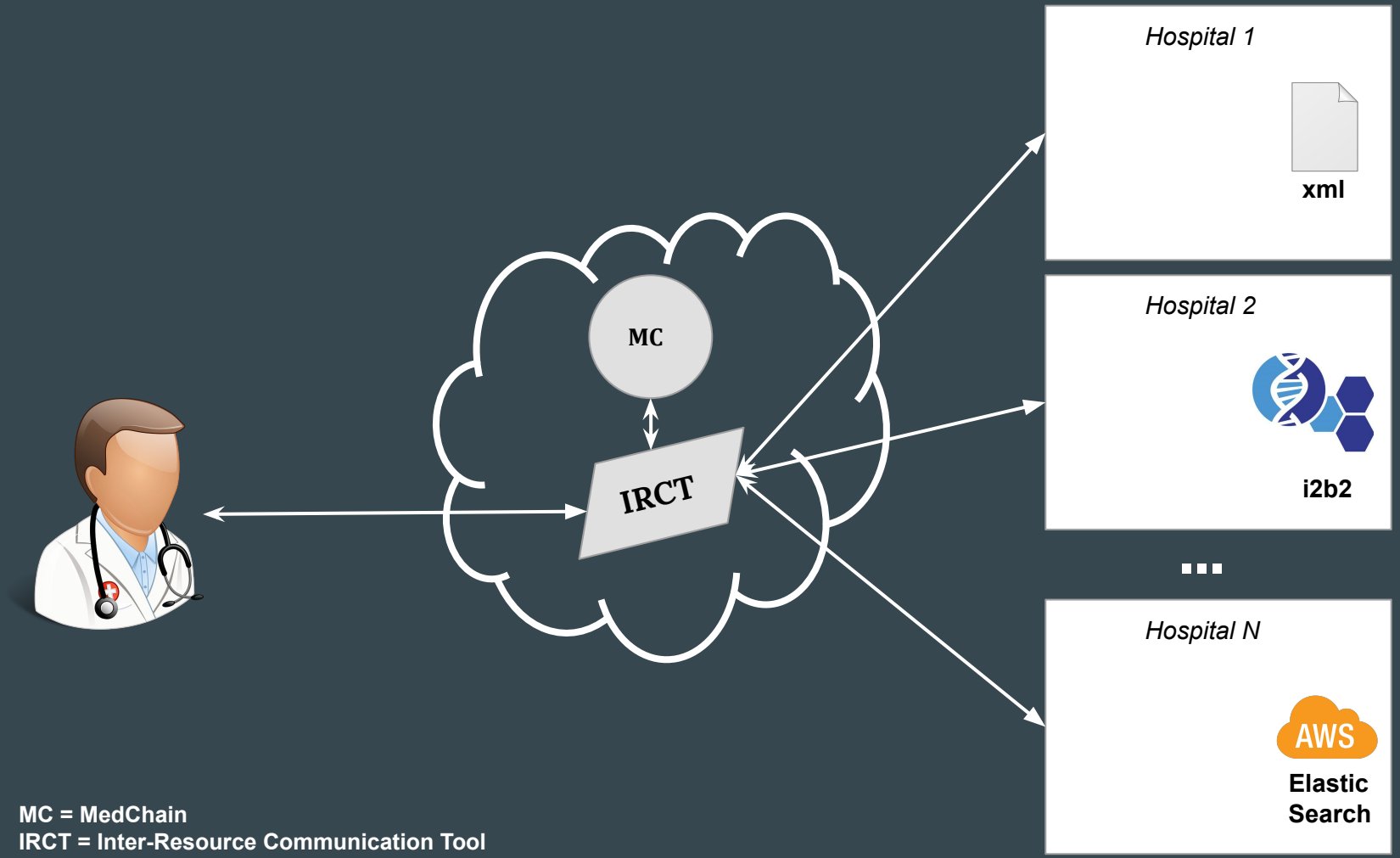
IdP = Identity Provider
IRCT = Inter-Resource Communication Tool



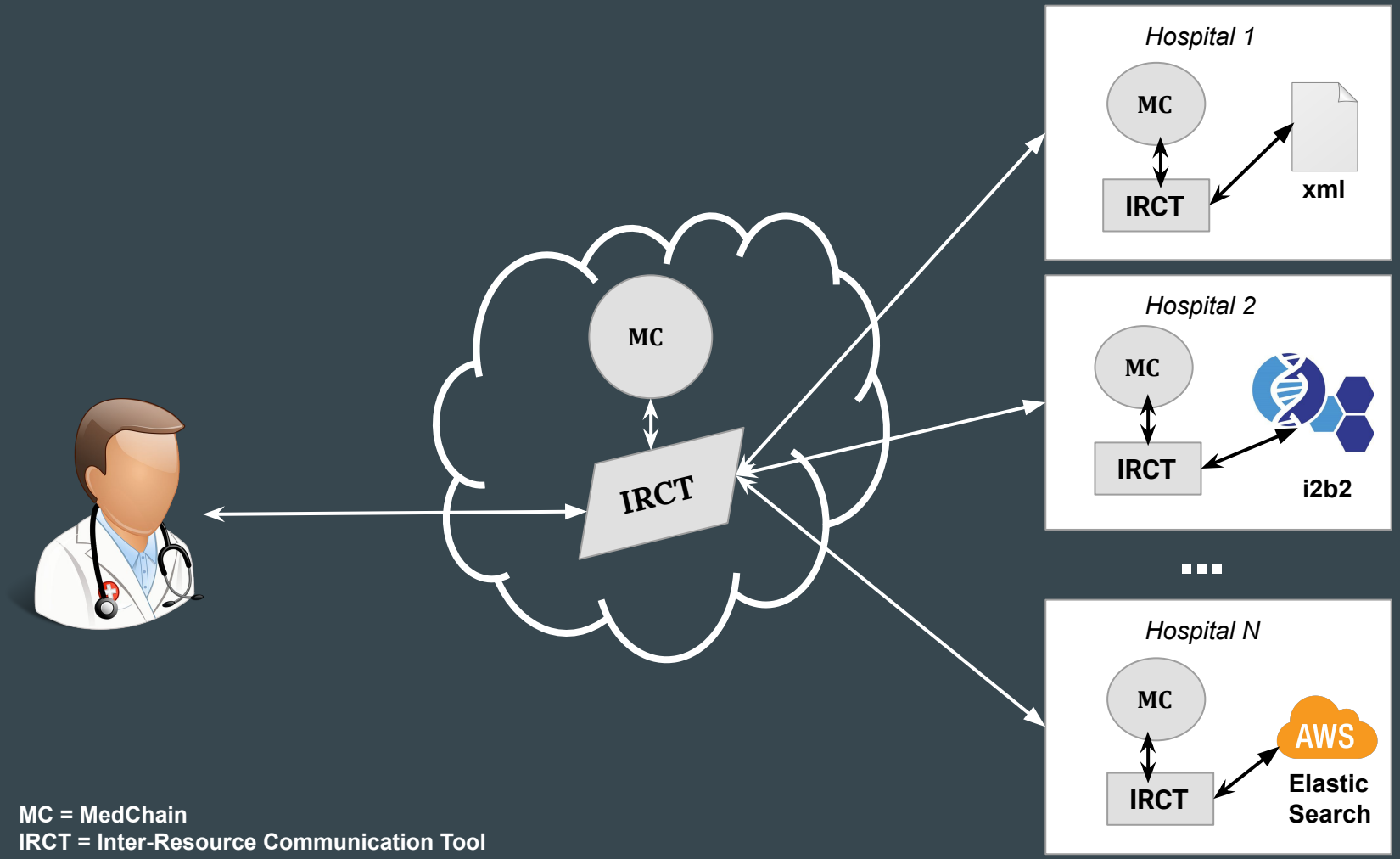
IRCT = Inter-Resource Communication Tool



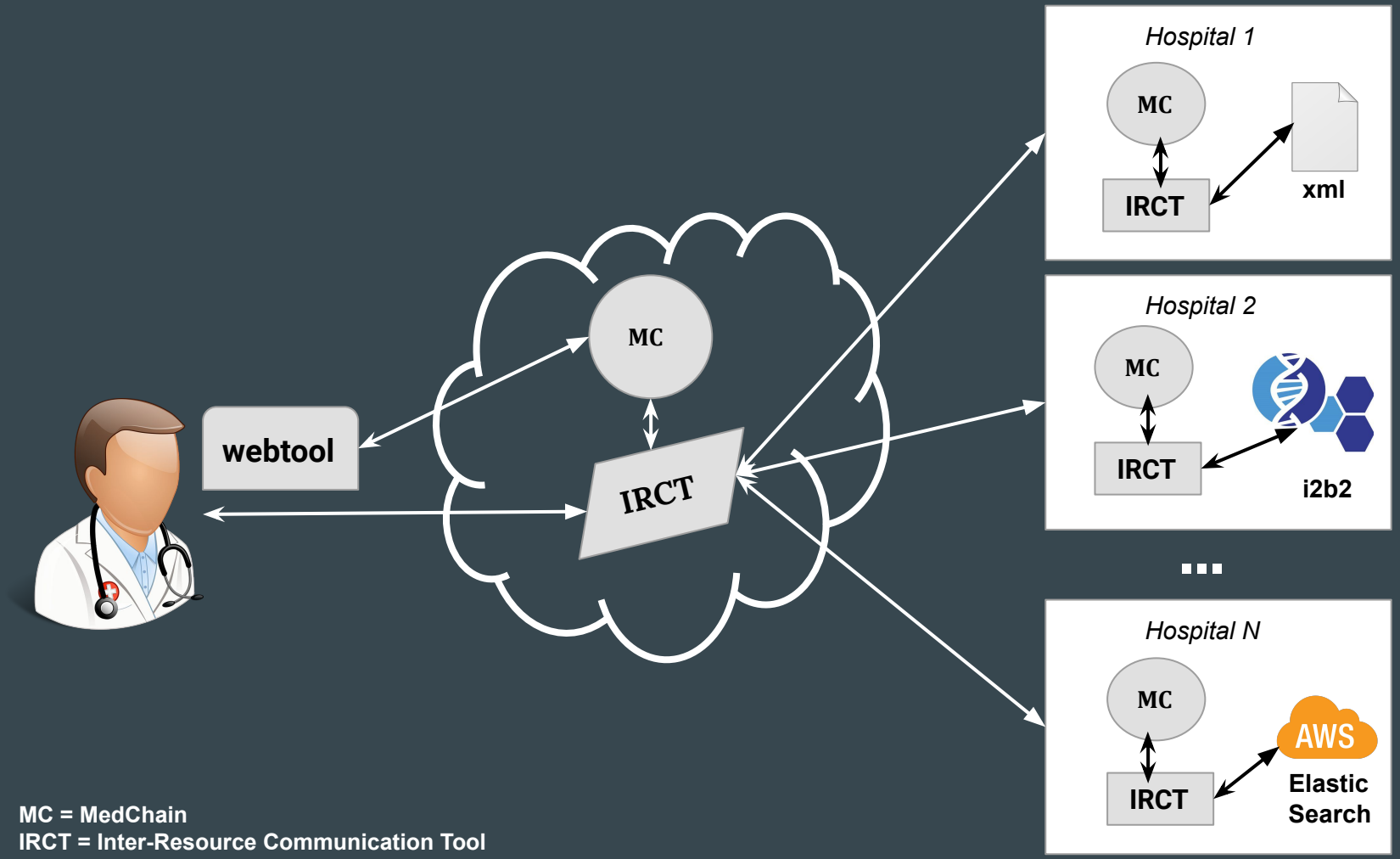
MC = MedChain
IRCT = Inter-Resource Communication Tool



MC = MedChain
IRCT = Inter-Resource Communication Tool



MC = MedChain
IRCT = Inter-Resource Communication Tool



MC = MedChain
IRCT = Inter-Resource Communication Tool

MedChain Features

Decentralization	• Requires ⅔ of the nodes to vote on changes • Control can be delegated to multiple people

MedChain Features

Decentralization	• Requires ⅔ of the nodes to vote on changes • Control can be delegated to multiple people
Auditability	• All accesses are logged on the blockchain

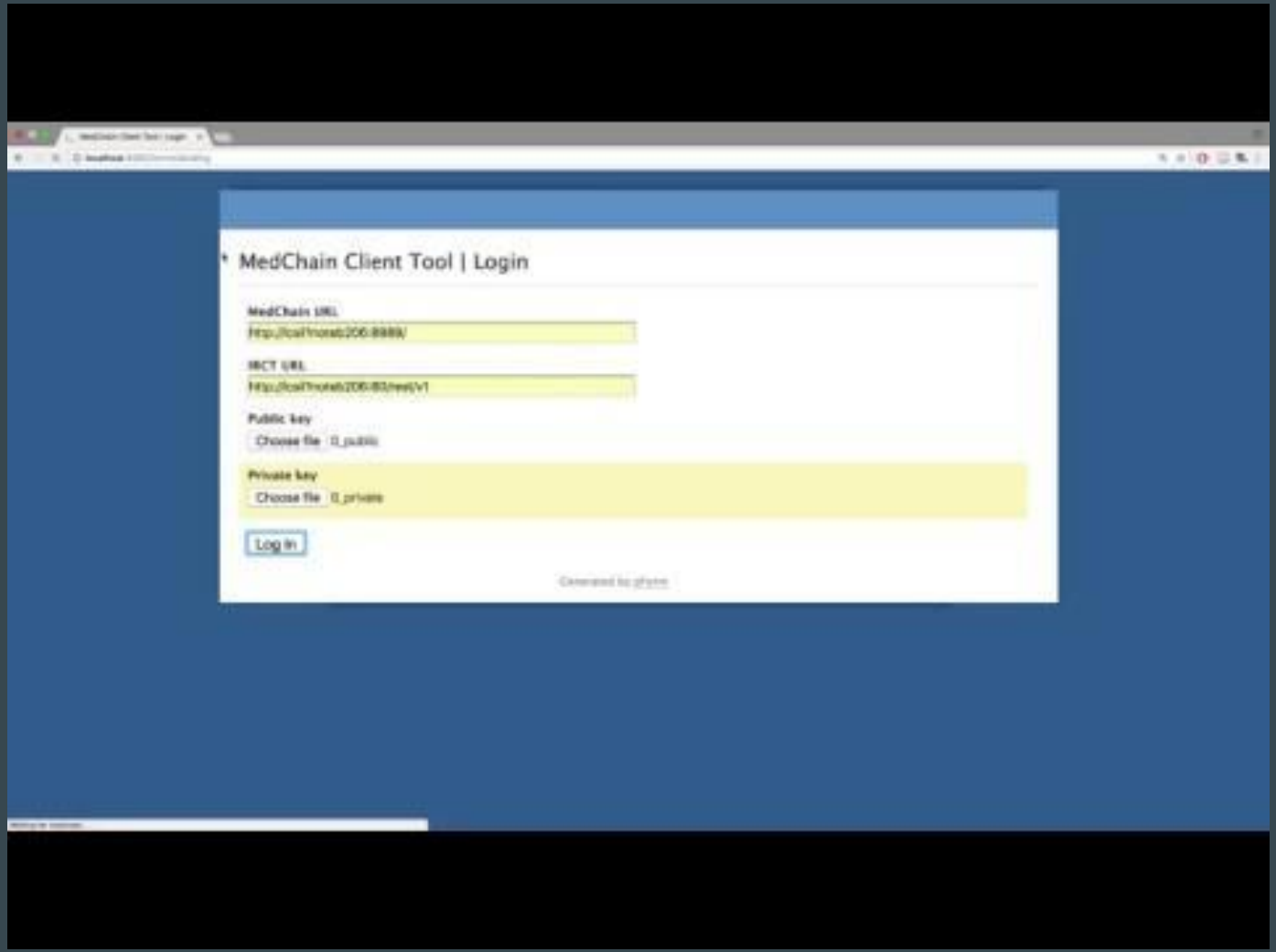
MedChain Features

Decentralization	• Requires ⅔ of the nodes to vote on changes • Control can be delegated to multiple people
Auditability	• All accesses are logged on the blockchain
Authorization	• Access control rules are stored on the blockchain, and enforced at the resources

MedChain Features

Decentralization	• Requires ⅔ of the nodes to vote on changes • Control can be delegated to multiple people
Auditability	• All accesses are logged on the blockchain
Authorization	• Access control rules are stored on the blockchain, and enforced at the resources
Rotation of keys	• Users can rotate their secret keys with ease, by evolving the respective DARCs

MedChain: Demo*


MedChain Client Tool | Login
MedChain URL
IRCT URL
Public key
Choose file
Private key
Choose file
Log In

MedChain: Performance Evaluation

Experimental Setup

- 3 servers (from the IC Cluster) to form a cothority for MedChain

Experimental Setup

- 3 servers (from the IC Cluster) to form a cothority for MedChain
- **Network conditions**
  - artificial delay of 15ms (mean) and 10ms (std. deviation) on packets from each server.

Experimental Setup

- 3 servers (from the IC Cluster) to form a cothority for MedChain
- Network conditions
  - artificial delay of 15ms (mean) and 10ms (std. deviation) on packets from each server.
- **Experiments with increasing number of users / projects to see how well the system scales**

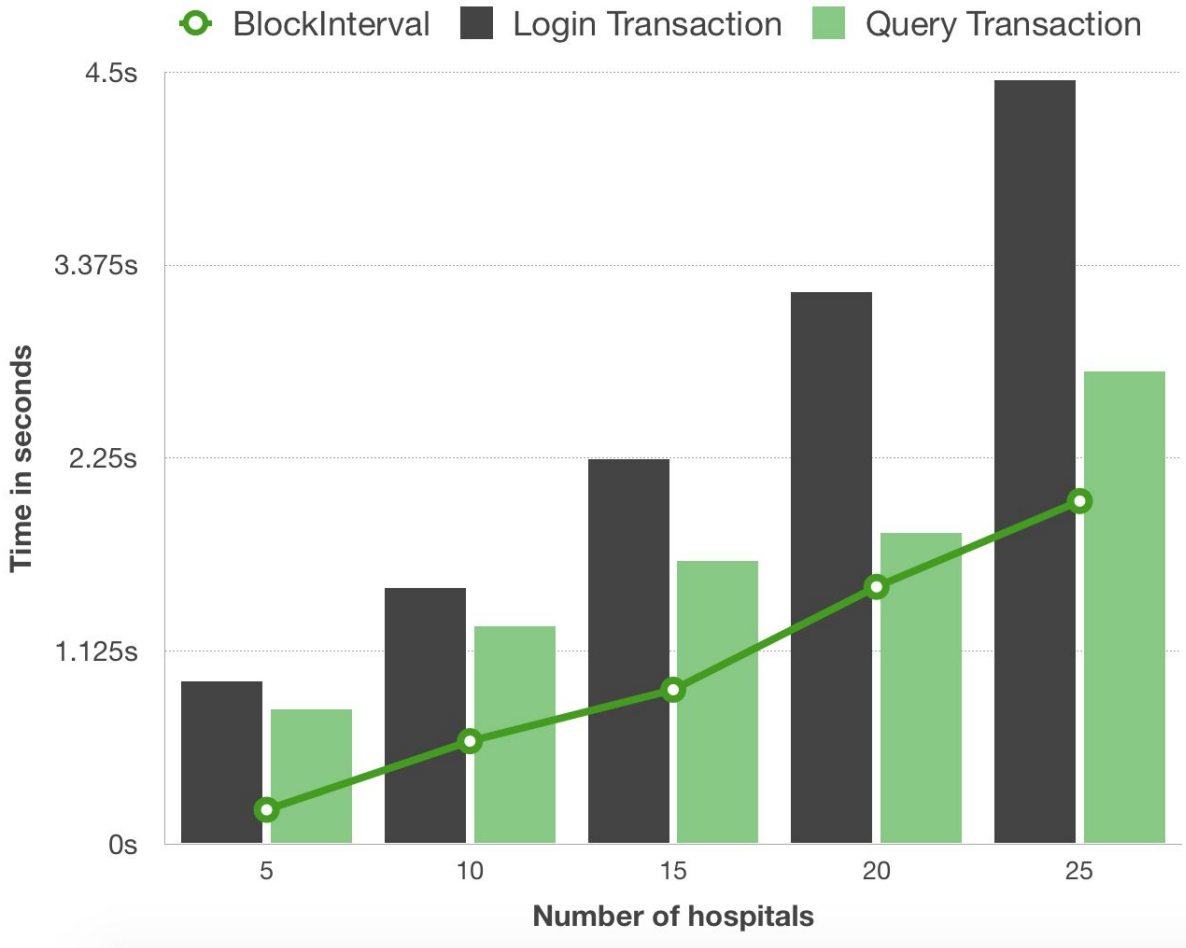
Experimental Setup

- 3 servers (from the IC Cluster) to form a cothority for MedChain
- Network conditions
  - artificial delay of 15ms (mean) and 10ms (std. deviation) on packets from each server.
- Experiments with increasing number of users / projects to see how well the system scales
- **Each data-point in the results corresponds to the median from 5 runs of an experiment**

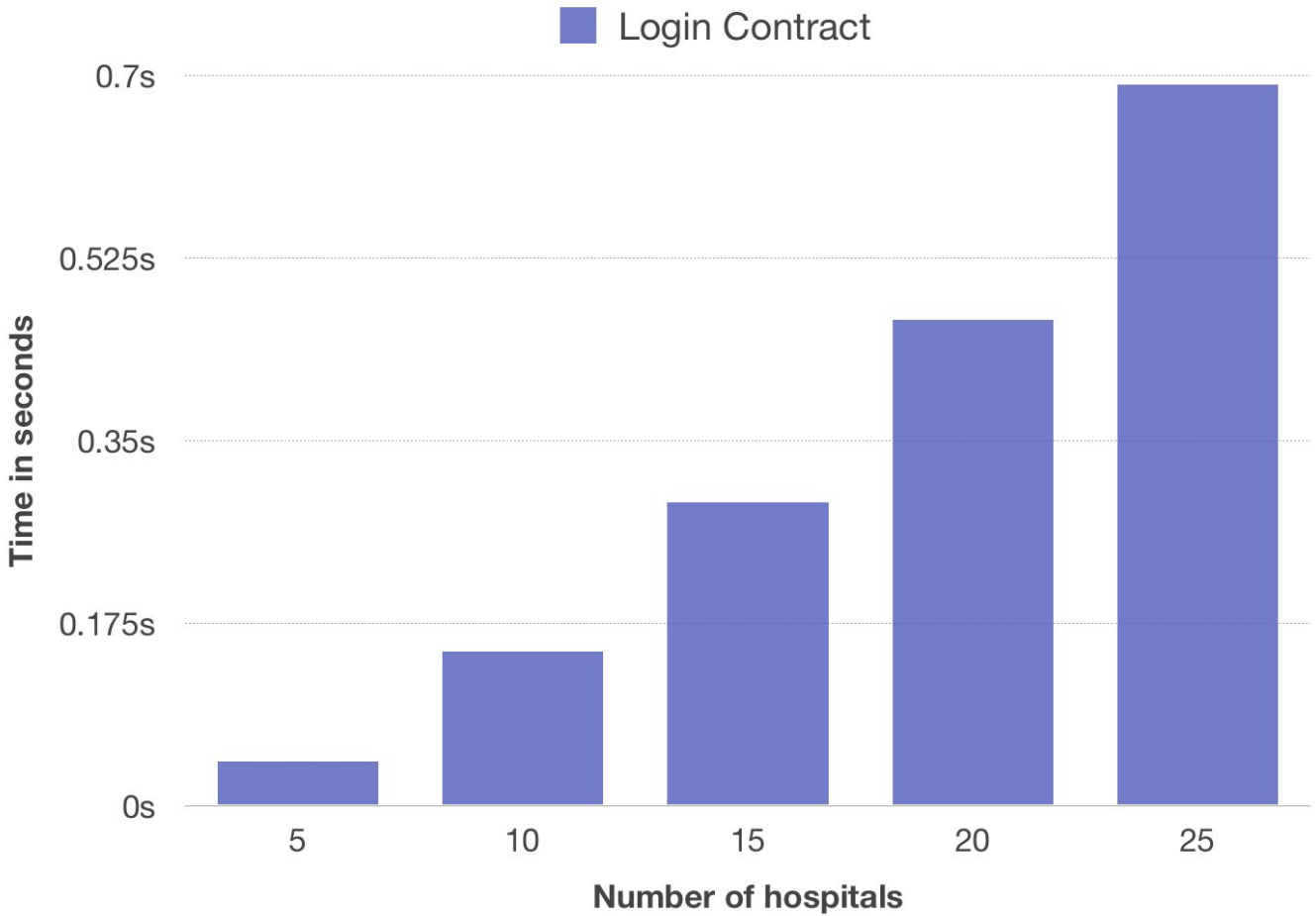
Experiment 1: System Overview

- **Time taken** vs **number of hospitals** in the system
- Other variables were configured as follows
  - **50 users / hospital**
  - **3 projects / hospital**
  - **20% hospitals / project**
- Block-interval was fine-tuned for each run

Time taken during user-login, and **query appendment**



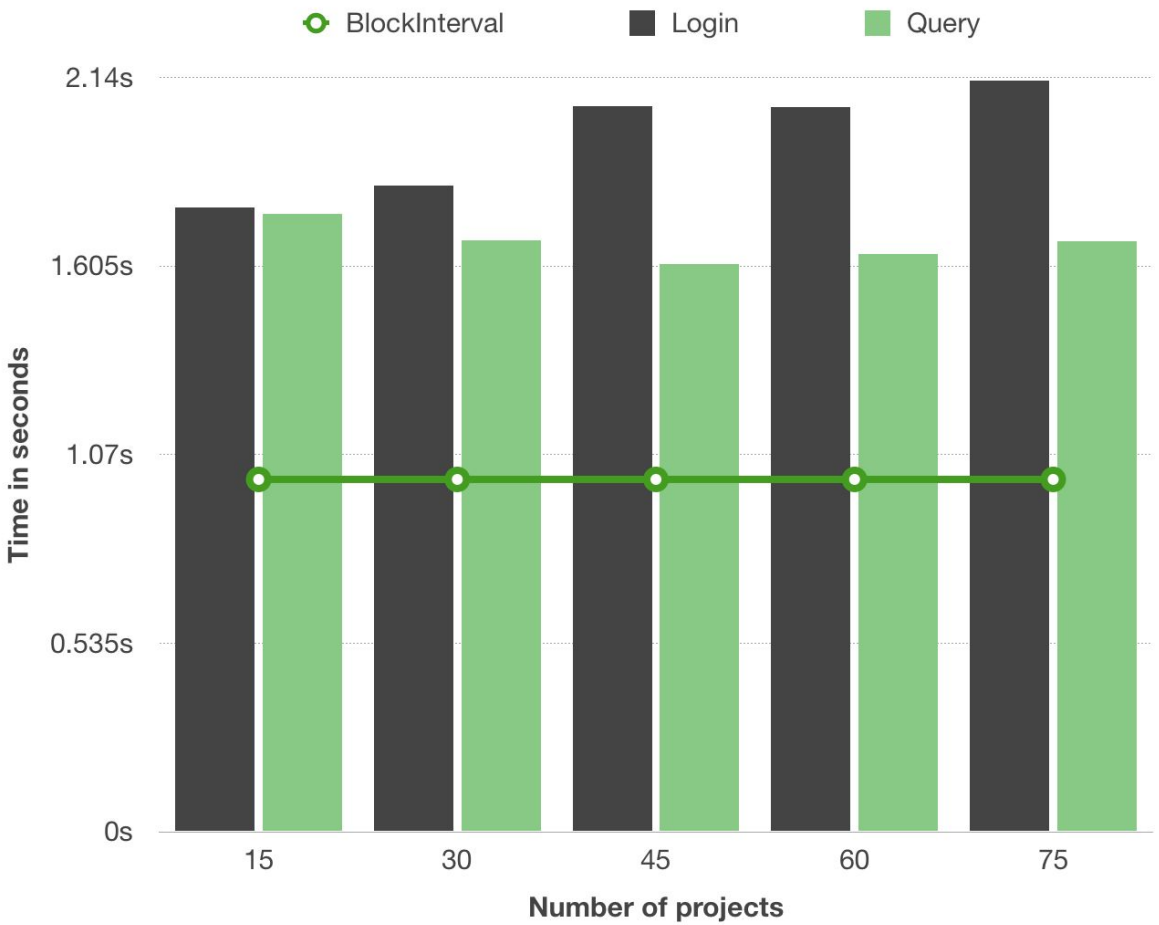
Time taken by the login contract at each node



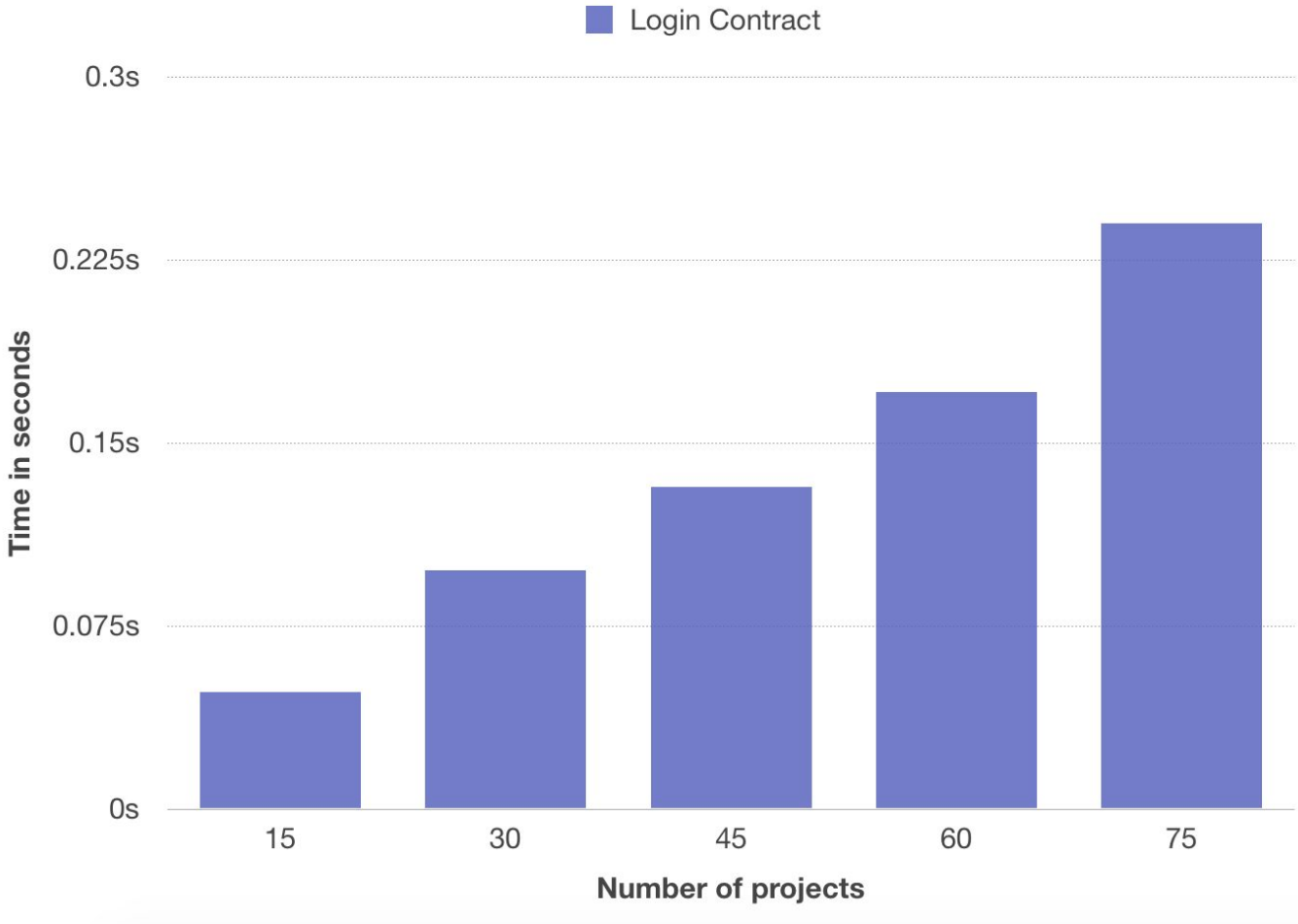
Experiment 2(a): Differential Analysis

- **Time taken** vs **number of projects** in the system
- Other factors constant:
  - **15 hospitals**
  - **750 users**
  - **150 users / project**
- Block-interval was kept constant at 1 second

Time taken during user-login, and **query appendment**



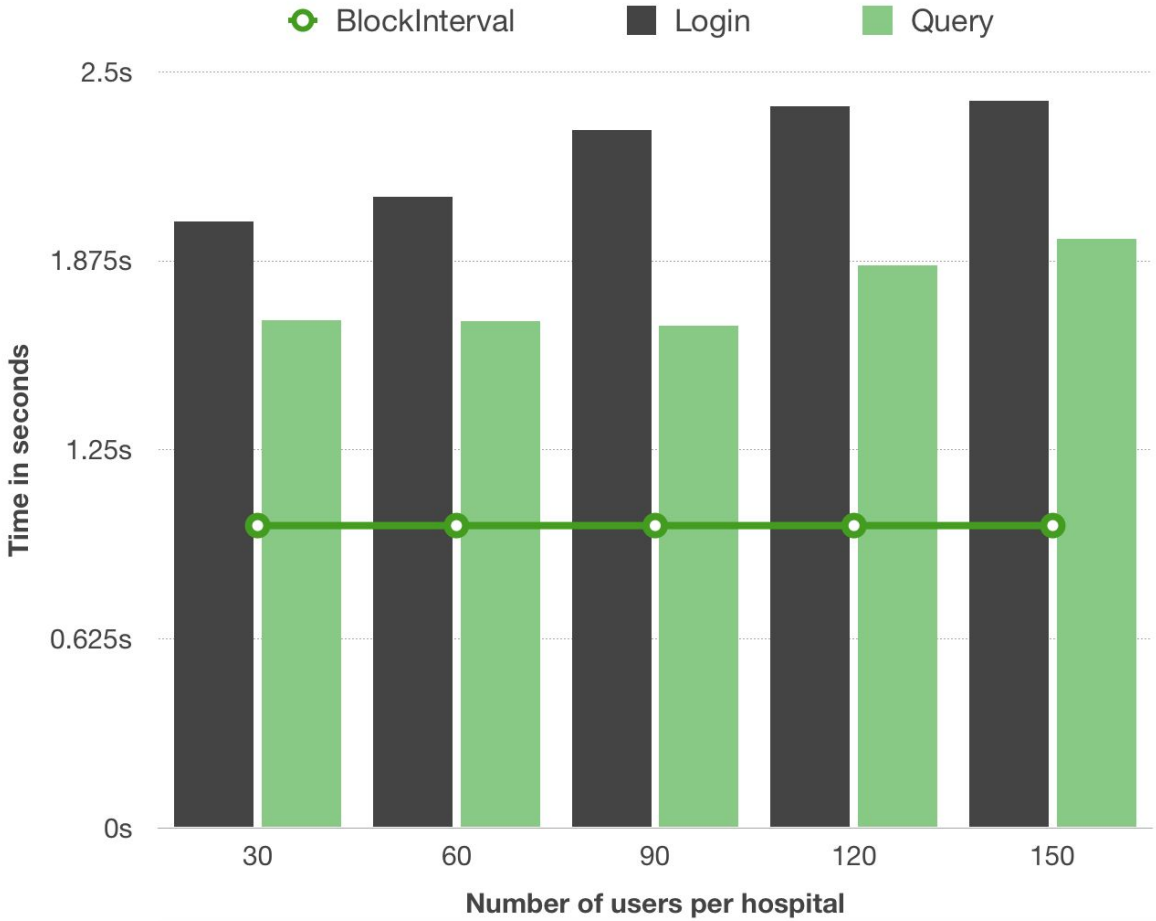
Time taken by the login contract at each node



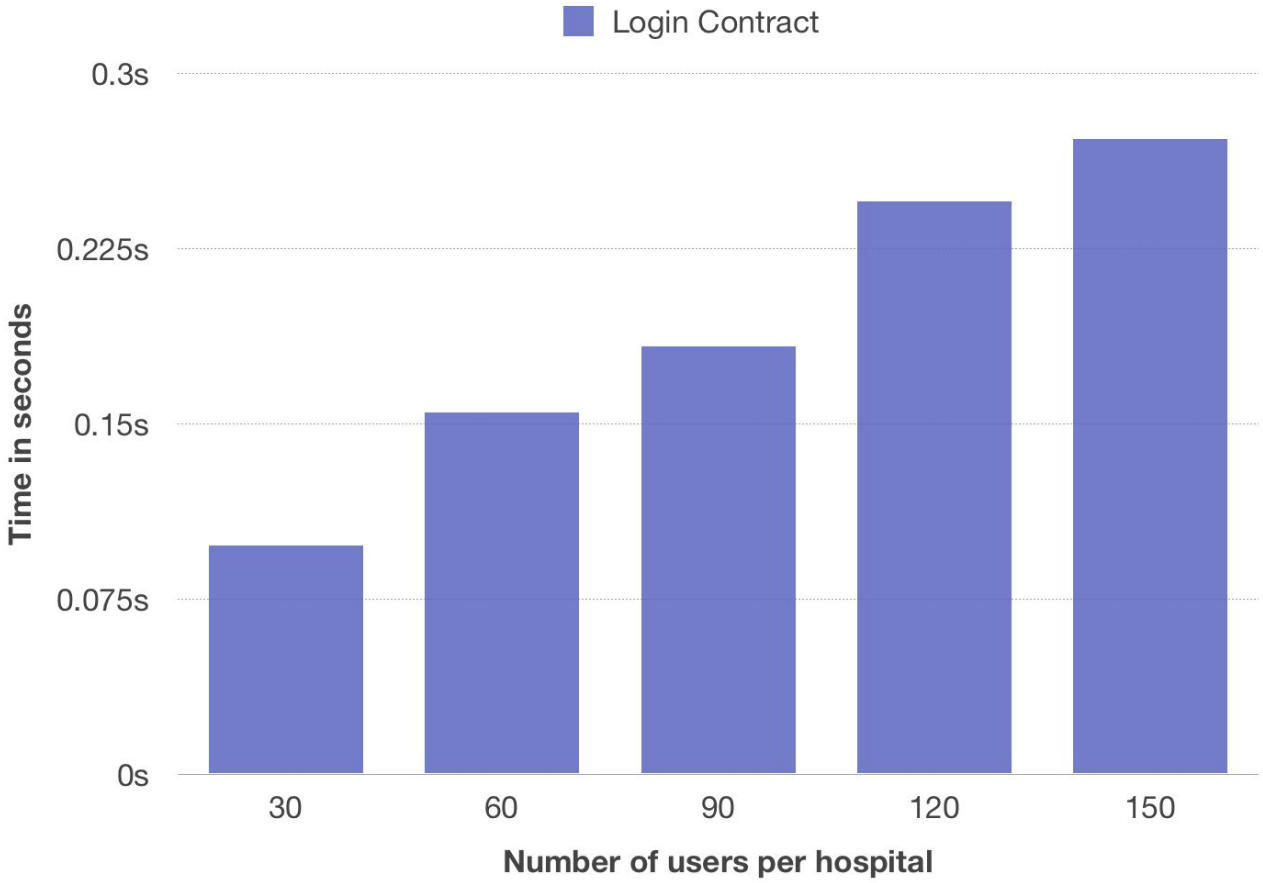
Experiment 2(b): Differential Analysis

- **Time taken** vs **number of users/hospital** in the system
- Other factors constant:
  - **15 hospitals**
  - **45 projects**
  - **3 hospitals / project**
- Block-interval was kept constant at 1 second

Time taken during user-login, and **query appendment**



Time taken by the login contract at each node



Conclusion

Contributions

- Developed MedChain by writing contracts (and DARCs) on top of Omniledger.

Contributions

- Developed MedChain by writing contracts (and DARCs) on top of Omniledger.
- **Designed the system architecture**

Contributions

- Developed MedChain by writing contracts (and DARCs) on top of Omniledger.
- Designed the system architecture
- **Modified the IRCT tool**
  - **to validate user login through MedChain**
  - **to validate queries through MedChain**

Contributions

- Developed MedChain by writing contracts (and DARCs) on top of Omniledger.
- Designed the system architecture
- Modified the IRCT tool
  - to validate user login through MedChain
  - to validate queries through MedChain
- **Developed tool for end-users to interact with the MedChain**

Contributions

- Developed MedChain by writing contracts (and DARCs) on top of Omniledger.
- Designed the system architecture
- Modified the IRCT tool
  - to validate user login through MedChain
  - to validate queries through MedChain
- Developed tool for end-users to interact with the MedChain

All code + documentation available at https://github.com/DPPH/MedChain & https://github.com/DPPH/IRCT

Future Work

- Integrate MedChain with the Glowingbear GUI for IRCT

Future Work

- Integrate MedChain with the Glowingbear GUI for IRCT
- **Develop an interface for managers / administrators of the hospitals**

Future Work

- Integrate MedChain with the Glowingbear GUI for IRCT
- Develop an interface for managers / administrators of the hospitals
- **Do more extensive performance evaluation**

Future Work

- Integrate MedChain with the Glowingbear GUI for IRCT
- Develop an interface for managers / administrators of the hospitals
- Do more extensive performance evaluation
- **Logging / confidentiality of results via MedCo integration**

Final Remarks

- Have had an excellent summer

Final Remarks

- Have had an excellent summer
- **Thanks to the supervisors**

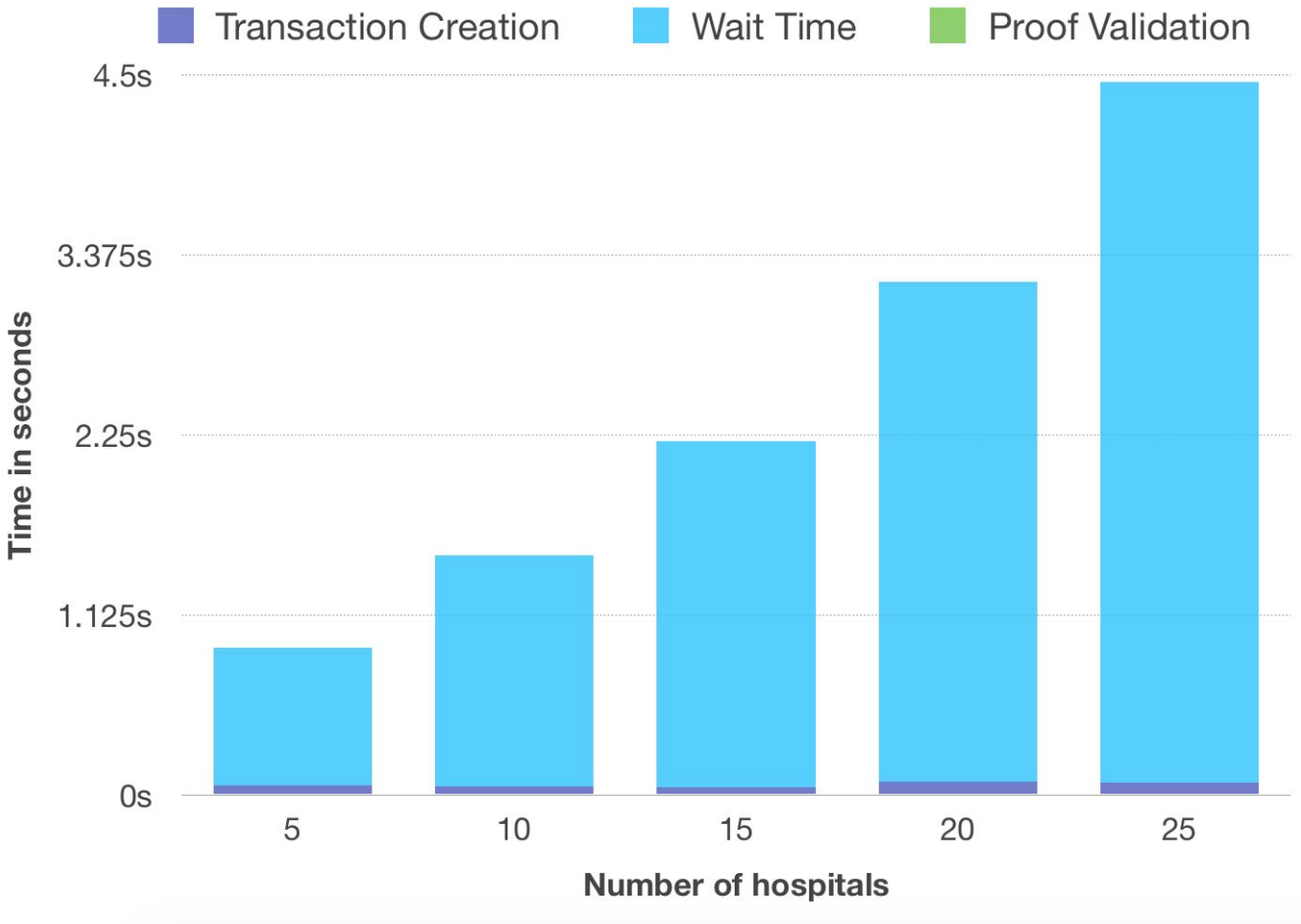
Final Remarks

- Have had an excellent summer
- Thanks to the supervisors
- **Thanks to collaborators from the DEDIS lab**

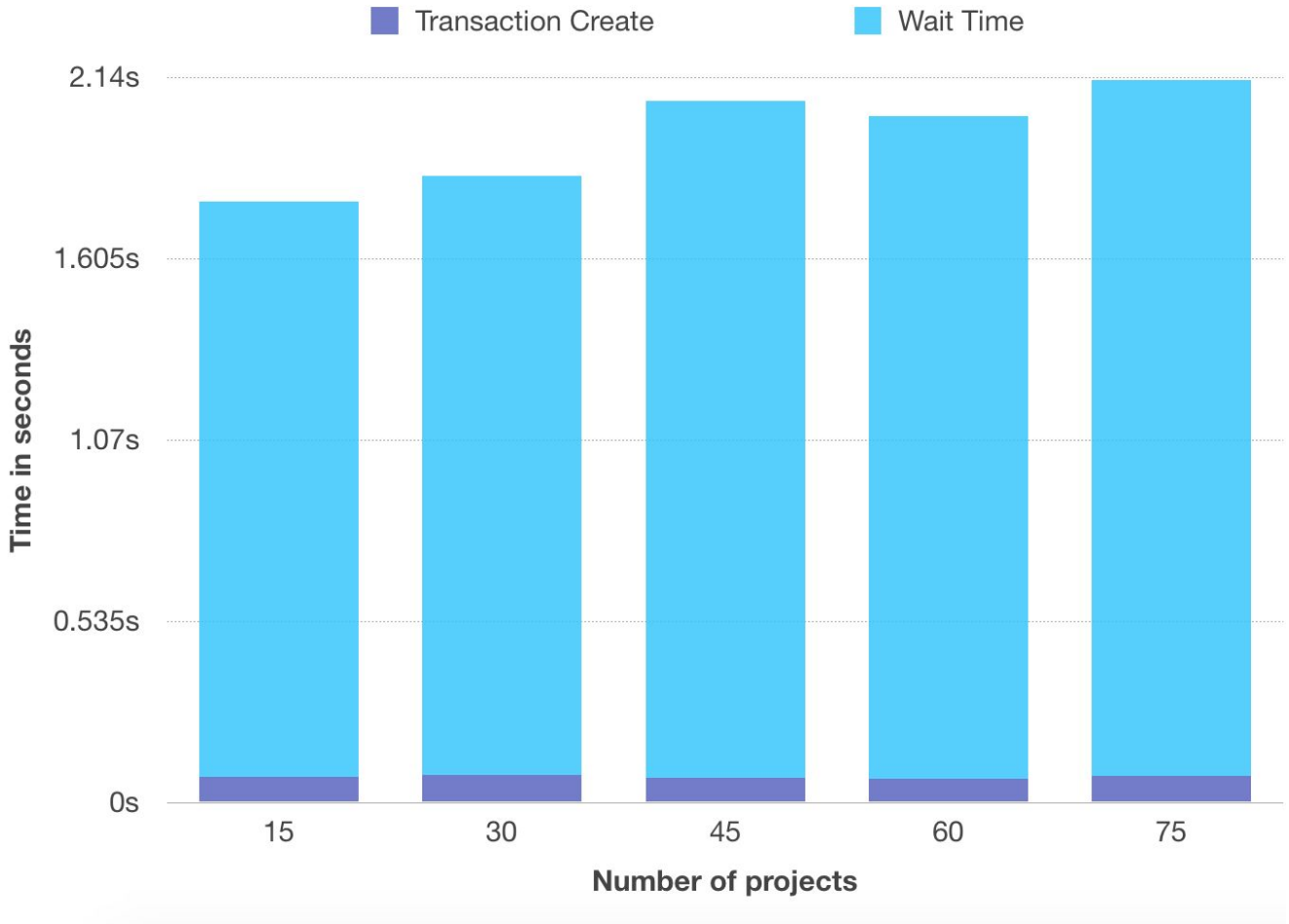
Thank you for listening.

Questions?

Breakdown of the time taken during user-login



Breakdown of the time taken during user-login



Breakdown of the time taken during user-login