FortiRx: Distributed Ledger based Verifiable and Trustworthy Electronic Prescription Sharing

Presenter: Anand Kumar Bapatla

Anand Kumar Bapatla[1], S. P. Mohanty[2], E. Kougianos[3]

University of North Texas, Denton, TX, USA.[1,2,3]

Email: **ab0841@unt.edu[1], saraju.mohanty@unt.edu[2], elias.kougianos@unt.edu[3]**

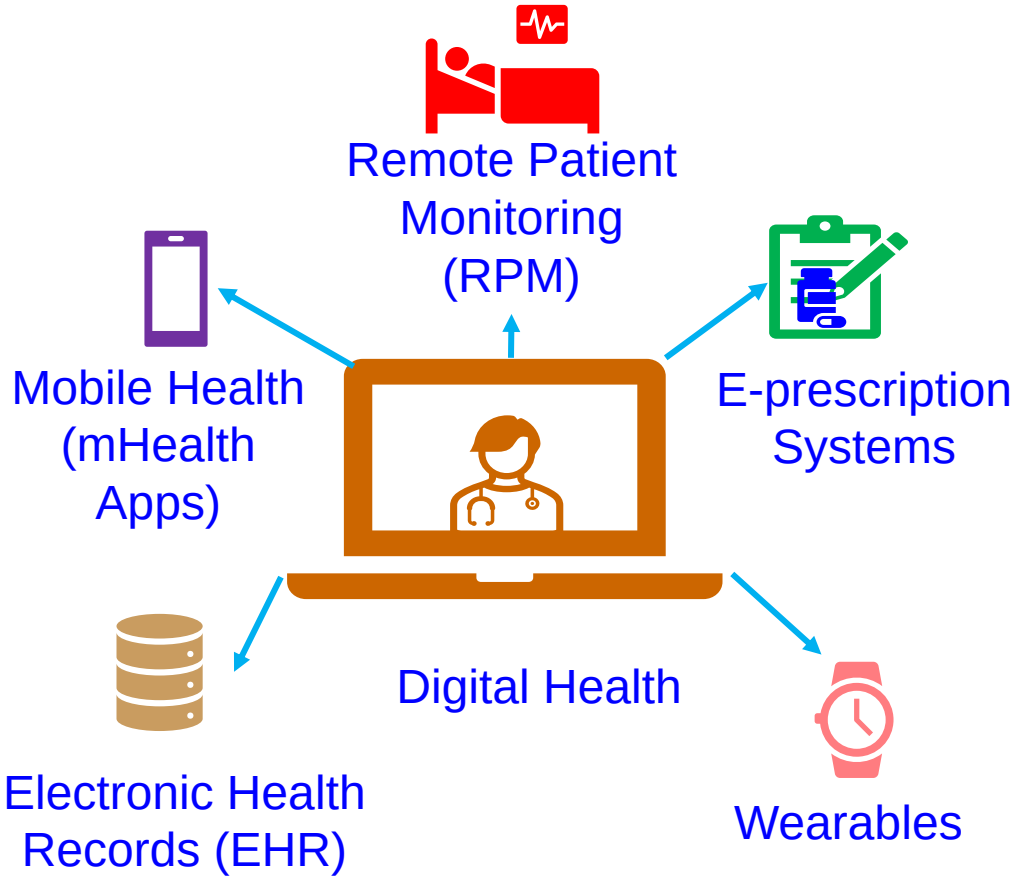
Outline

- Digital Health Technologies and E-Prescription
- Challenges
- Blockchain as a Solution
- Novel Contributions
- Architectural Overview
- Implementation Details
- Results and Analysis
- Conclusion

Digital Health Technologies and E-Prescription

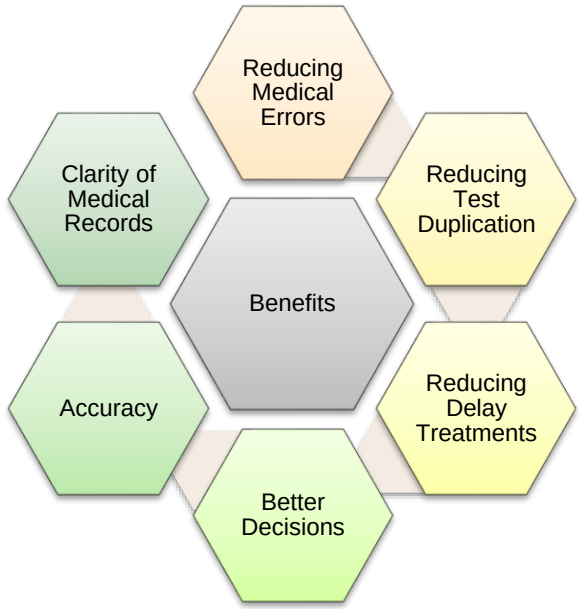
What are Digital Health Technologies?

- Digital Health Technologies encompasses a range of digital tools and platforms to improve healthcare services
- Facilitates remote consultation, personalized health tracking, and data-driven interventions
- E-prescription systems are crucial components of Digital Health Technologies and are often integrated into Electronic Health Records



Electronic Health Records (EHR's)

- Electronic Health Record (EHR) is an electronic version of patient medical history maintained by the provider
- Contains demographics, progress notes, problems, medications, and other administrative information



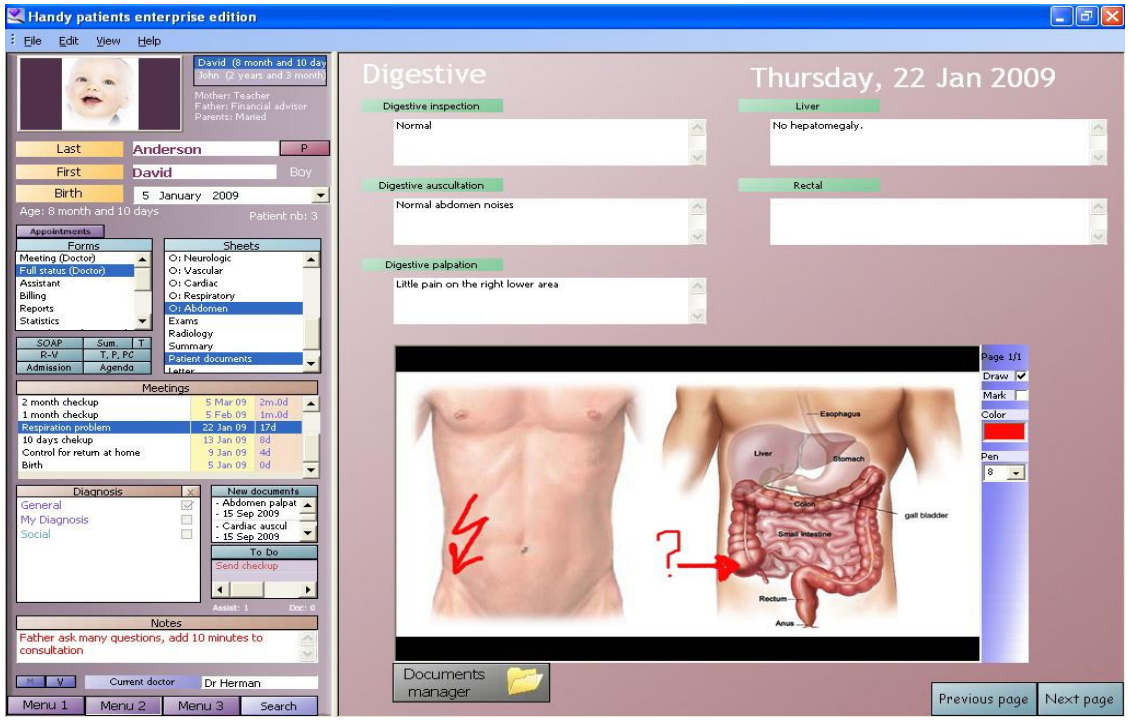
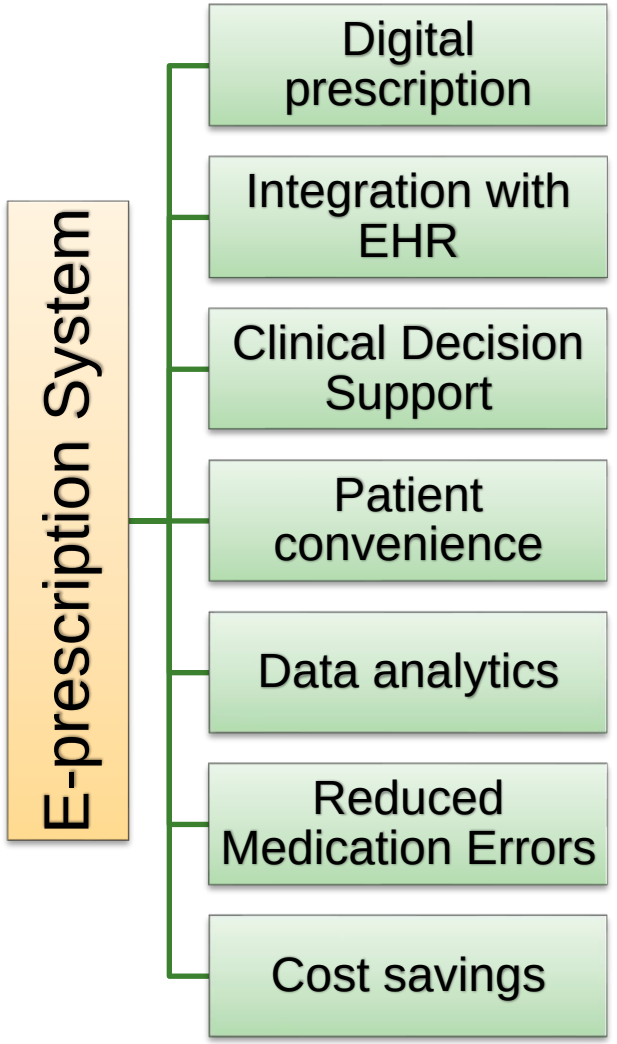


Image Source: DaCarpenther, An electronic medical record example, Handy patients electronic medical record (free open-source version)

Electronic Prescription

- Revolutionized the way medications are prescribed, processed, and dispensed
- Digital version of prescriptions increase legibility and reduces medication errors
- Clinical Decision Support Tools – Warn potential drug interactions, suggest alternate medication, offer dosage recommendations

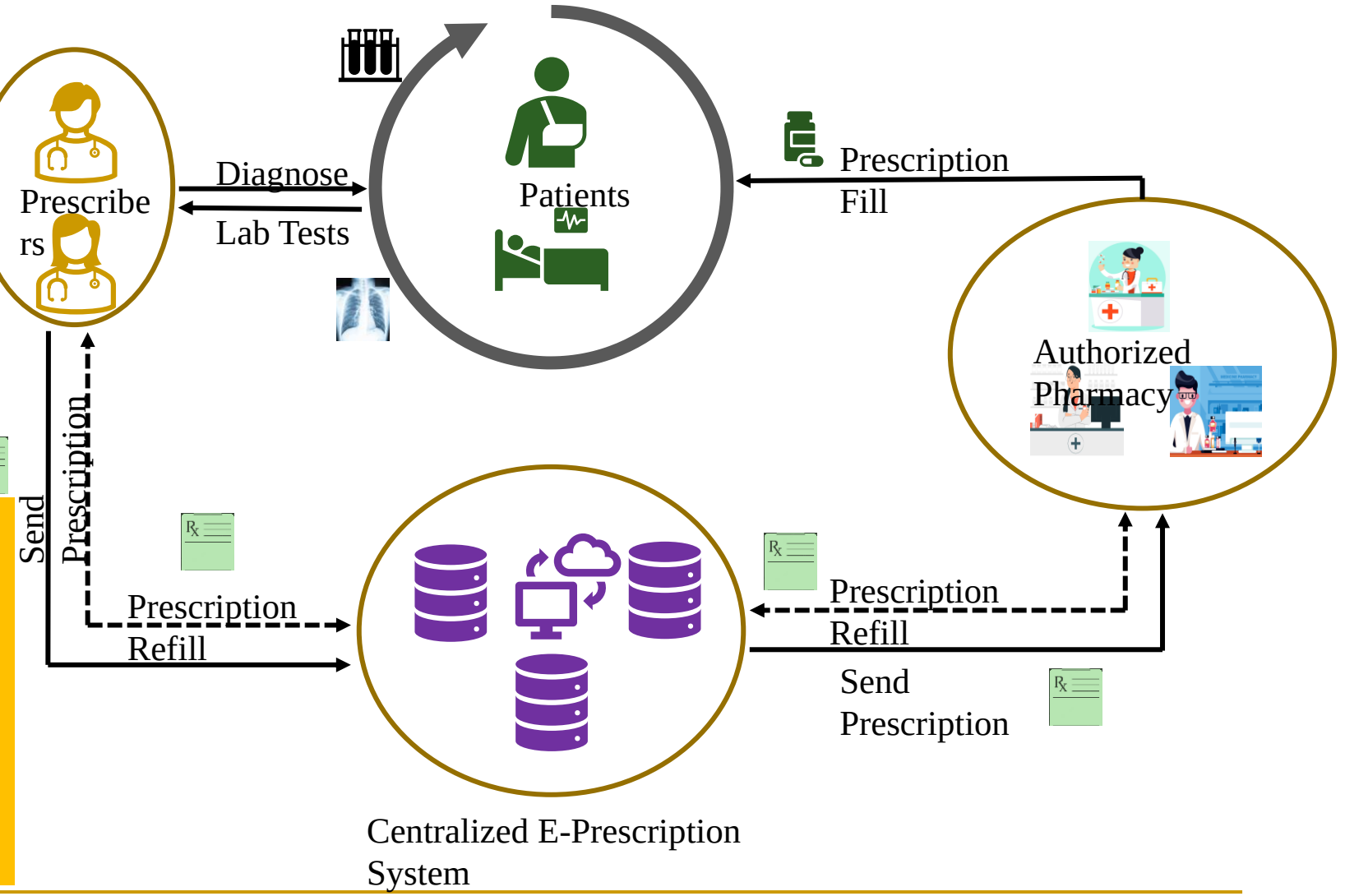
- More than 100,000 reports of medication errors (FDA)
- 40% of Americans report being involved in medical errors (Institute for Healthcare Improvement/NORC at the University of Chicago)
- 1 in 5 doses of medication provided during patient visits is administered incorrectly



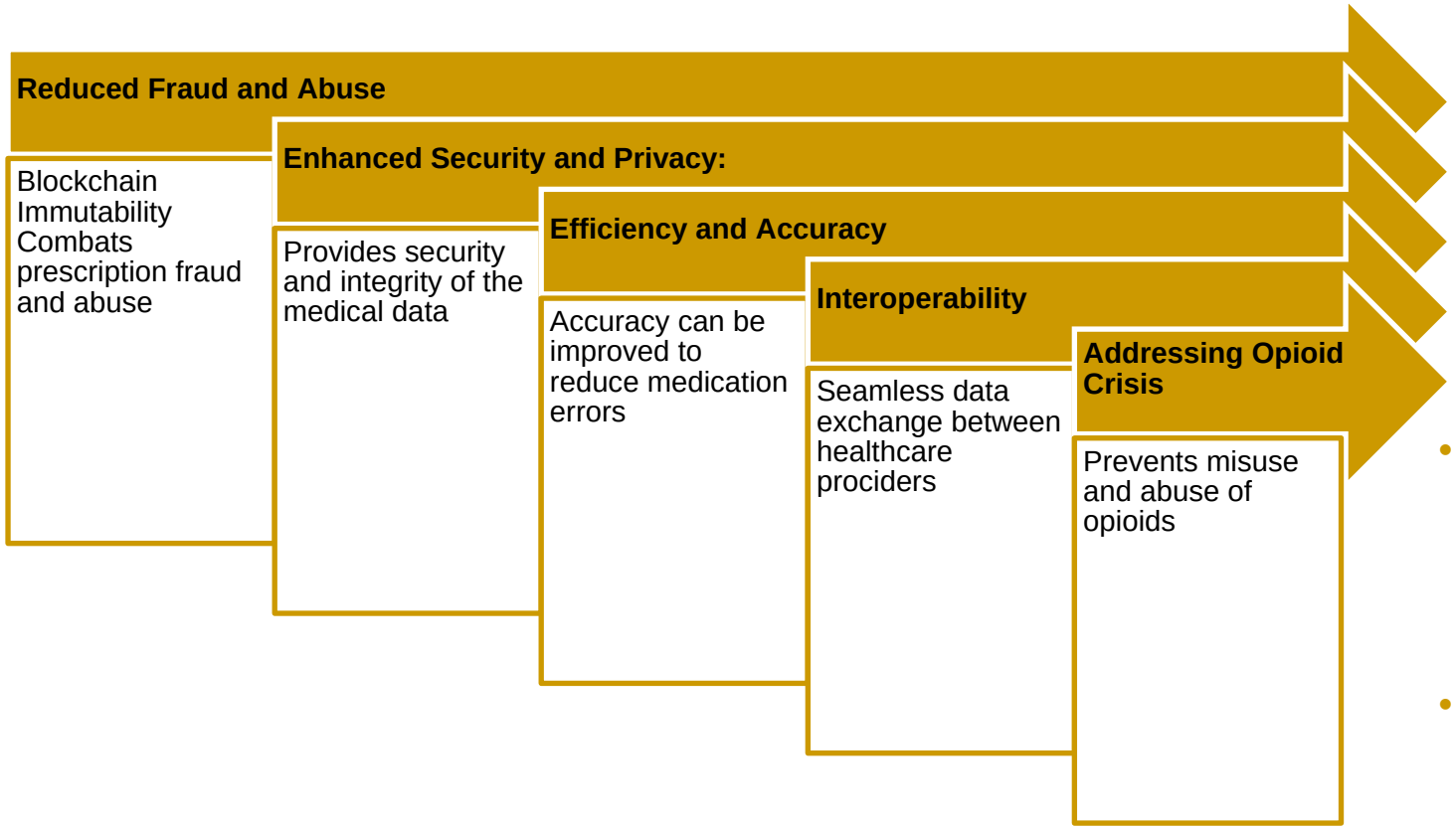
E-Prescription System and Issues

- Single Point of Failure (SPOF)
- Data Security
- Privacy Concerns
- Interoperability Concerns (PDMP)
- System availability Issues

- Prescription Drug Monitoring Programs(PDMP) help mitigate prescription misuse and diversion
- Oversight of controlled substance prescriptions



Motivation

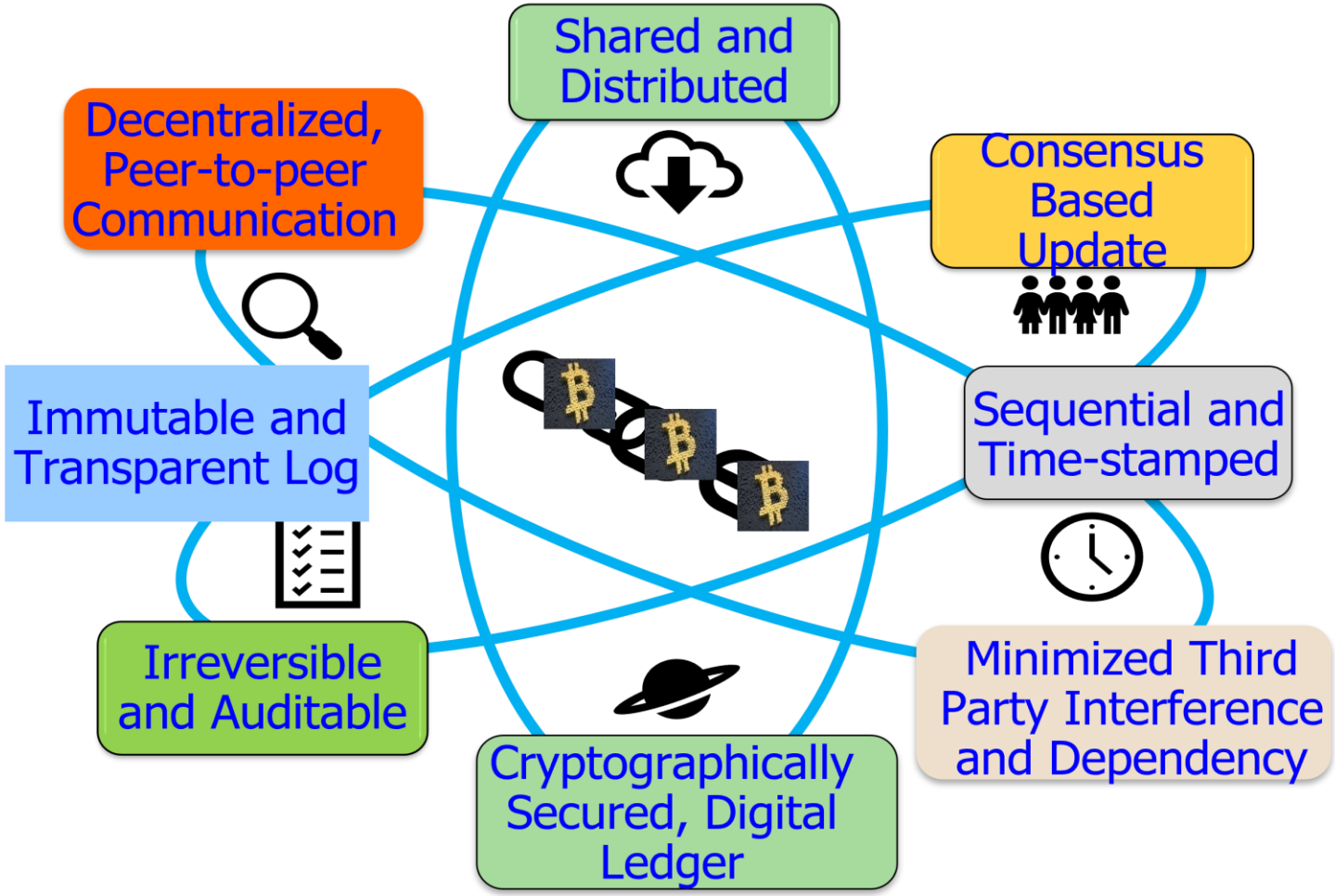


Prescription Drug Type	Annual Abusers	% Among Rx Abusers	% Among Americans
Painkillers	9.7 million	59.5%	3.43%
Opioids Alone	9.3 million	57.1%	3.29%
Sedatives	5.9 million	36.2%	2.08%
Stimulants	4.9 million	30.1%	1.73%
Benzodiazepine Alone	4.8 million	29.4%	1.70%
All Prescription Drugs	16.3 million	100%	5.76%

- 16M – 6% of Americans over the age of 12 abuse prescriptions in a year.
- 2M – 12% of prescription drug abusers are addicted.

Statistics Source: https://drugabusestatistics.org/prescription-drug-abuse-statistics/

Blockchain Technology



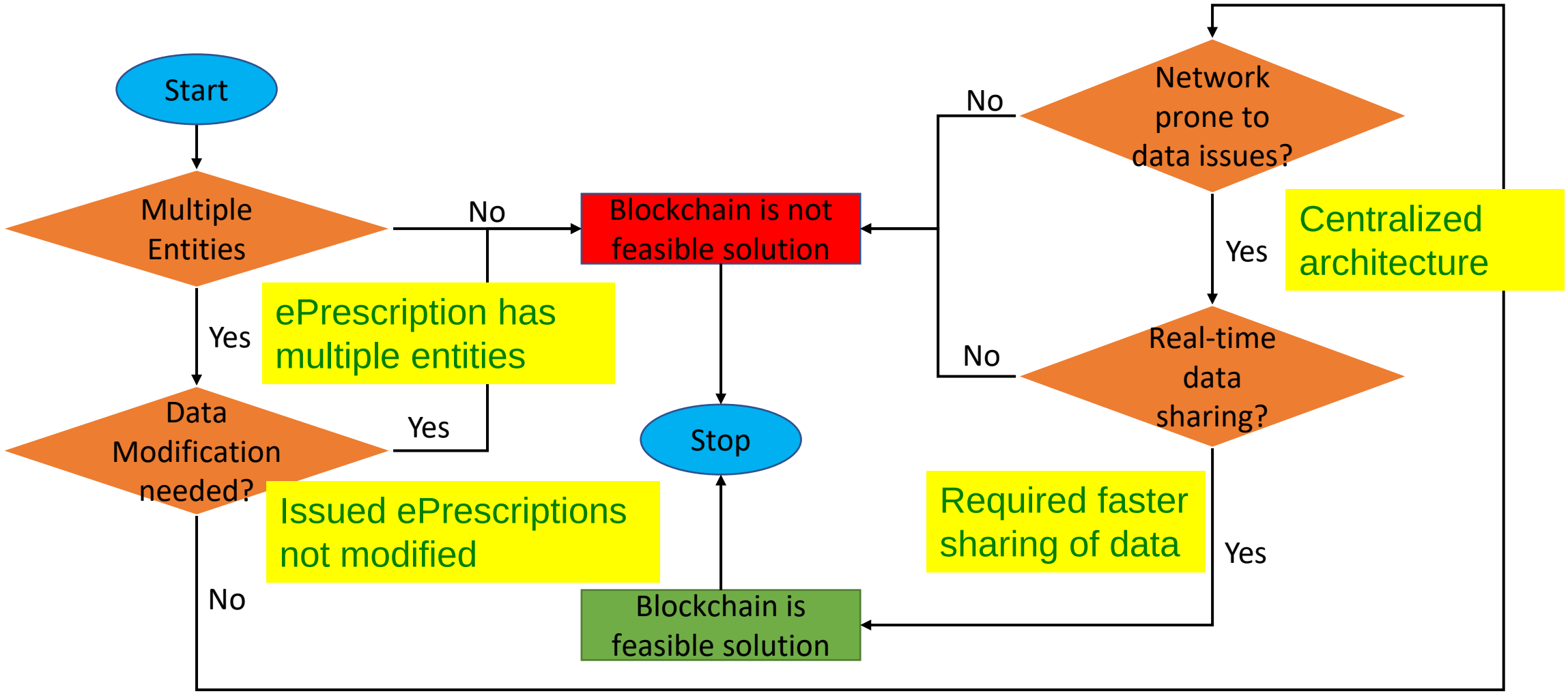
Technical Definition: A blockchain is a linked list that is built with hash pointers instead of regular pointers.
Socio-Political–Economic Definition: A blockchain is an open, borderless, decentralized, public, trustless, permissionless, immutable record of transactions.
Financial – Accounting Definition: A blockchain is a public, distributed ledger of peer-to-peer transactions.

Source: D. Puthal, N. Malik, S. P. Mohanty, E. Kougianos, and C. Yang, "The Blockchain as a Decentralized Security Framework", *IEEE Consumer Electronics Magazine (CEM)*, Volume 7, Issue 2, March 2018, pp. 18--21.

Blockchain as a Solution

- Enhanced Data Security: Decentralized and immutable ledger reduces the risk of data breaches and maintains data integrity.
- Patient-Centric Privacy: Empowers patients to have control over their health data.
- Interoperability: Improves interoperability between healthcare providers, pharmacies, PDMP databases, and other participants of the prescription process.
- High Availability: Blockchain-based e-prescription systems are more resilient to downtime, ensuring uninterrupted access.
- Automated Processes: Smart Contracts can automate various aspects of the e-prescription process.

Evaluating Blockchain for E-prescription



Novel Contributions

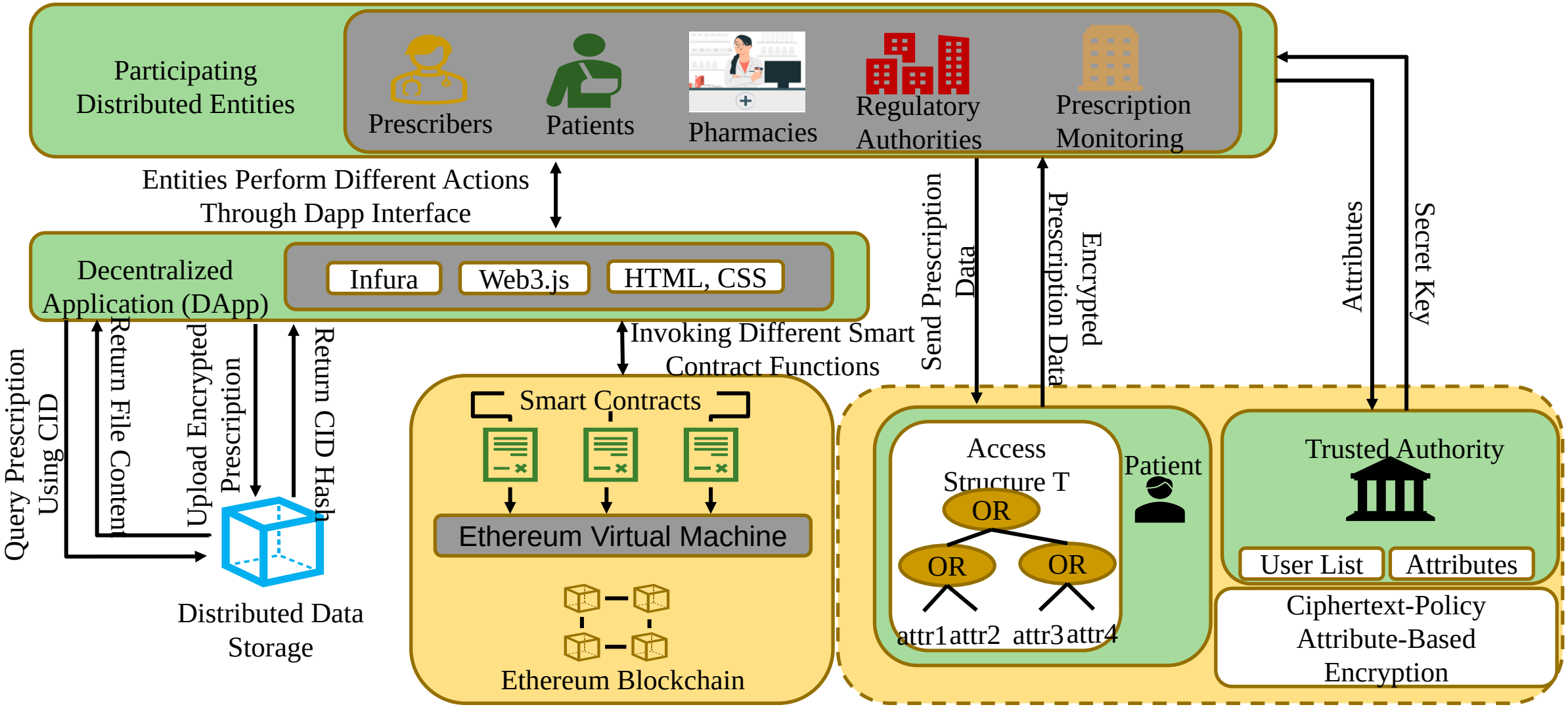
- Proposed FortiRx makes use of blockchain combined with the distributed file system (IPFS) to create a decentralized environment for all the participating entities.
- Blockchain enhances the interoperability of the system.
- Usage of off-chain distributed file-sharing systems to store prescription information can help in reducing the amount of on-chain data.
- It is resistant to Single Point of Failure (SPOF) and reduces response latency
- It avoids data tampering and prescription abuse
- Cipher text-policy attribute-based encryption (CP-ABE) provides a robust access control mechanism.

Comparative Analysis with State-of-Art

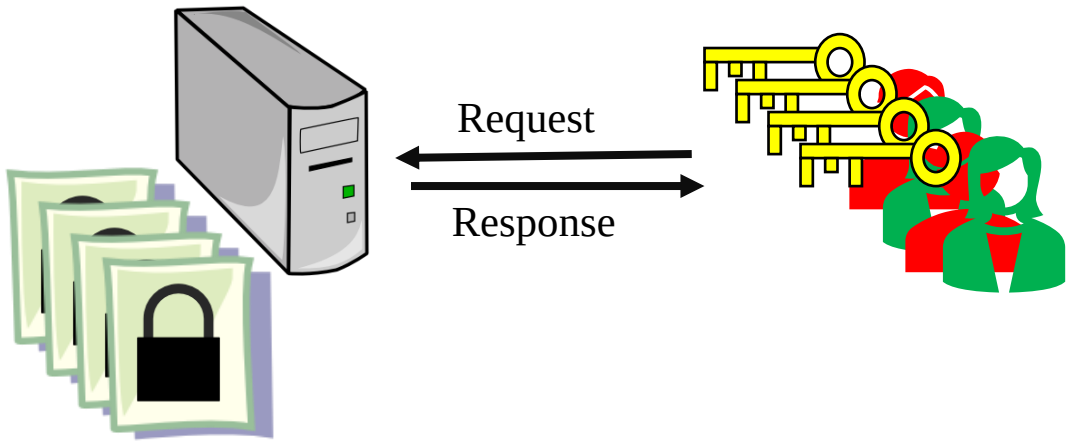
	Blockchain Platform	Smart Contracts	Off-chain storage	Data Privacy	Access Control Mechanism	CP-ABE
Thatcher, et al. 2018 [20]	Ethereum	✓	✗	✗	✗	✗
Musamih, et al. 2021 [14]	Ethereum	✓	✓	✗	✓	✗
Taylor, et al. 2022 [19]	Ethereum	✓	✗	✓	✓	✗
Aluaimi, et al. 2022 [3]	Ethereum	✓	✓	✗	✓	✗
Ionescu, et al. 2022 [11]	Ethereum	✓	✗	✗	✗	✗
FortiRx (Current Paper)	Ethereum	✓	✓	✓	✓	✓

Architecture

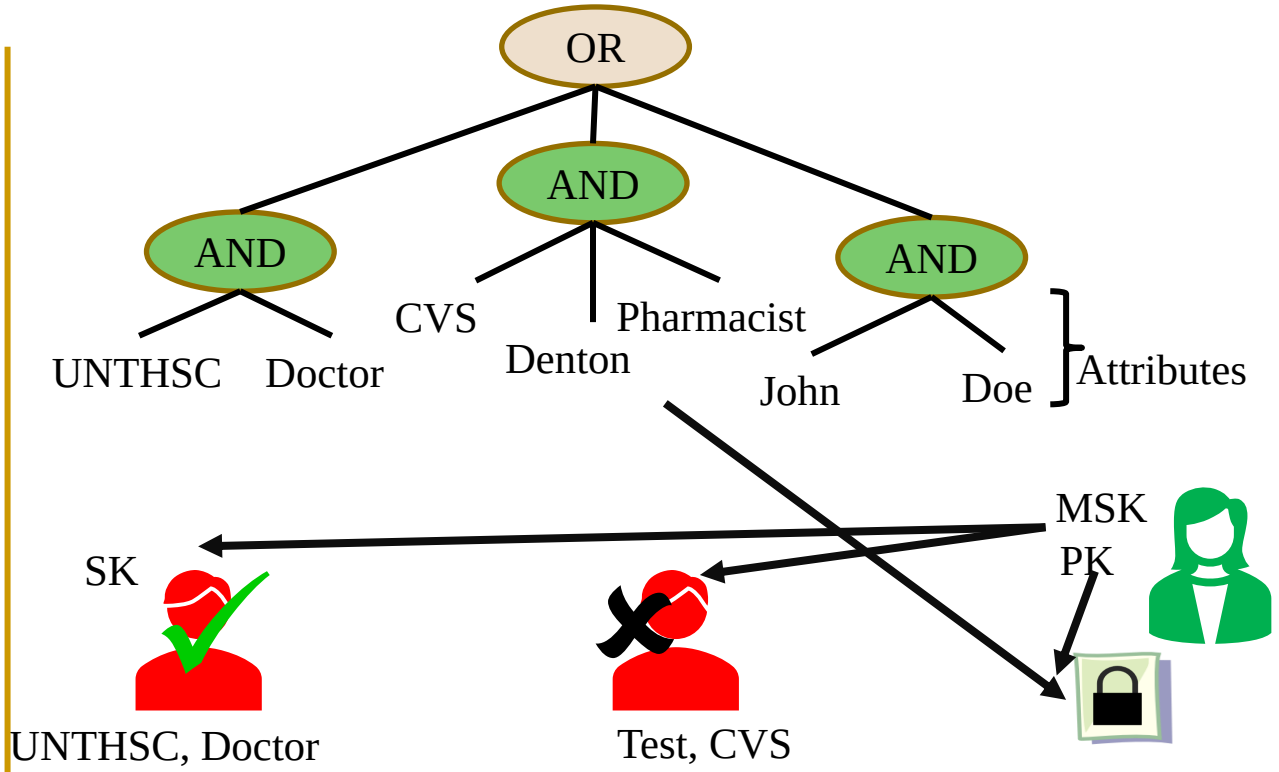
FortiRx Architecture



Asymmetric Encryption vs CP-ABE



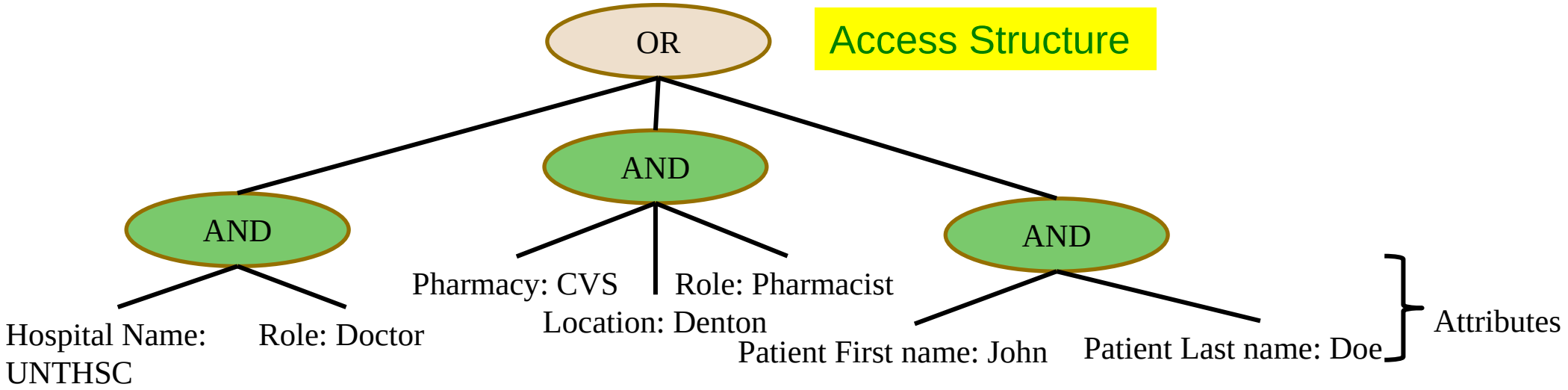
- Secure but not flexible
- New key for every participant
- Fine-grain access control not possible
- Needs efficient key distribution



- User private keys based on “attributes”
- Files can be encrypted under “policy” over those attributes
- Can only decrypt if attributes satisfy policy

Access Control Mechanisms

- For Prescription Access
  - Cipher text-policy attribute-based encryption (CP-ABE) allowing fine-grained control of data access
  - Data-sharing among multiple parties without revealing the content of the data.
  - Access the data based on attributes, such as roles or clearances rather than specific keys.
  - Effectively scales as the number of parties involved in a multi-party access scenario grows



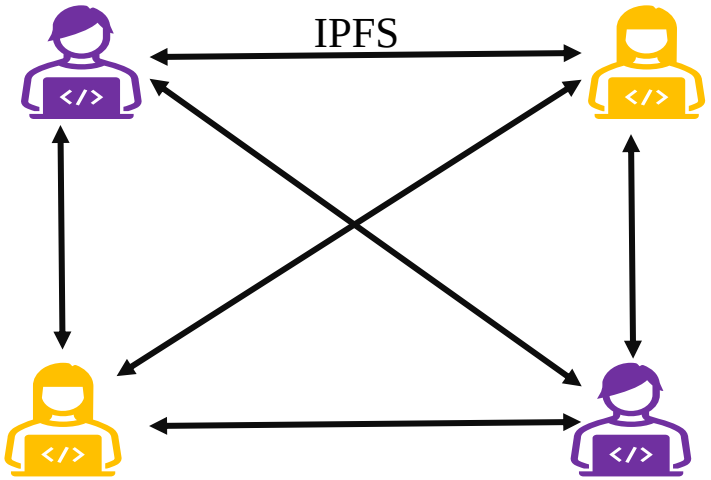
Access Policy = ((“Hospital Name”== “UNTHSC” AND “Role”== “Doctor”) OR (“Pharmacy”==“CVS” AND “Location”==“Denton” AND “Role”==“Pharmacist”) OR (“Patient First Name”==“John” AND “Patient Last Name”==“Doe”))

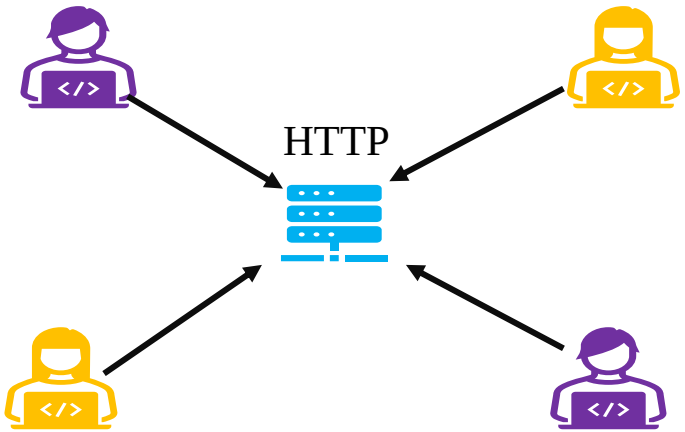
Access Control Mechanisms

- CP-ABE Steps
  - Key Generation – Generates a master key and a set of attributes
  - Attribute Assignment – Attributes assigned to users or entities
  - Policy Specification – The data owner specifies an access policy using a set of attributes
  - Encryption – The data owner encrypts using the access policy
  - Decryption – Requesting user attributes evaluated against policy and information before decrypting
- For Role Specific Functions
  - Role-Based Access Control (RBAC) mechanism automated using smart contracts.
  - Define, assign, and revoke roles for specific External Owner Accounts (EOA)
  - Modifiers defined and assigned to different smart contract functions
  - Authenticate role-based transactions and prevent unauthorized access.

Distributed Data Storage (IPFS)

Aspect	HTTP	IPFS
Adoption and Support	Widespread, universally supported	Growing adoption, expanding support
Protocol Complexity	Simple, well-established	Decentralized, content-addressed
Caching	Supports various caching mechanisms	Distributed content, local caching
Direct Access	Connects to centralized web servers	Peer-to-peer and distributed access
Control	Centralized control by web server	Decentralized, no single control
Data Addressing	URL-based addressing	Content-based addressing (hashes)
Redundancy and Resilience	Limited redundancy	Content distributed across nodes
Data Immutability	Can be updated by the server	Immutable content, cannot be changed
Data Sharing	Limited sharing without a server	Easy sharing with peers and nodes
Offline Access	Requires internet connection	Offline access to previously viewed
Data Retrieval Efficiency	Traditional DNS-based lookup	Efficient DHT-based content routing





Prescription Upload Steps in FortiRx

- Generate a digital prescription and create a file.
- For each prescription file, open and read the contents.
- Encrypt the prescription content using a public key and the patient's access policy, creating a ciphertext.
- For each encrypted prescription file and upload to IPFS.
- Retrieve the Content ID (CID) from the IPFS response.
- Create a new transaction in the prescription smart contract to create the prescription for the patient.
- If the caller of this transaction is the prescriber:
  - Create a new prescription and associate it with the patient's address.
  - Emit an event with prescription data, generating a log.
  - Return the transaction hash (Txhash).
- If the caller is not the prescriber, reject the transaction.

Algorithm 1 Proposed Prescription Upload Algorithm for FortiRx.

Input: Digital Prescription Data, public parameters (params,g1,g2,e) generated during CP-ABE setup, Access policy ρ defined by the patient
Output: Content ID for IPFS file, Transaction hash of prescription creation in blockchain

---

A digital prescription is generated, and a file is created
For each prescription file f do
    Open file in read mode
    FileItem ← open(filePath,'r')
    Read prescription content from the file
    prescription content (Pcontent) ← fileItem.read()
    Encryption is done using the public key (pk) and policy ρ to generate ciphertext of the prescription content
    Cipher text CT ← cpabe.encrypt(pk, Pcontent ,ρ)
    A new file is created and generated cipher text is written to that file
end for
For each encrypted prescription file f do
    Send upload request to IPFS
    Response (res) ← requests.post(Infura endpoint, authentication parameters, file f)
    Content ID from response is retrieved
    Content ID (CID) ← res.text['Hash']
end for
Prescriber creates a new createPrescription transaction in prescription smart contract
Transaction (Tx) ← prescription.createPrescription(patient address (Paddr),CID)
if caller == Prescriber then
    A new prescription is created and added to patient's address
    Emit an event (ev) with prescription data and a log is generated
    Return transaction hash (Tx hash)
else
    Reject Tx
end if

Prescription Retrieval Steps in FortiRx

- For each view request, Retrieve the prescription based on PID from the blockchain using the smart contract.
- Get the IPFS Hash (CID) of the prescription from the retrieved data.
- Request the prescription content from IPFS using CID.
- Receive the ciphertext (CT) from IPFS.
- Obtain a secret key for a specific set of attributes (attr list) from a trusted authority.
- Decrypt the ciphertext (CT) using the secret key to reveal the prescription content.
- Display the decrypted prescription content if the access policy (ρ) evaluates positively for the attribute list (attr list)
- If the access policy doesn't match the attribute list, decryption is not allowed.

Algorithm 2 Proposed Prescription Retrieval Algorithm for FortiRx.

Input: Prescription ID (PID) generated while creating new prescription in blockchain, attribute list of requesting entity (attr list)
Output: Decrypted prescription content (Pcontent)

1: For each view request (req) do
2: Send a function call to prescription smart contract to retrieve Prescription based on PID
3: Retrieved prescription Pret ← prescription.viewPrescription(PID)
4: Get IPFS Hash (CID) from the function response
5: CID ← Pret['IPFSHash']
6: Send a request to IPFS to retrieve prescription content (Pcontent)
7: Response (res) ← requests.post(Infura end point, CID, authentication parameters)
8: Retrieved cipher text (CT) ← res.text
9: Secret key for a set of attributes attr list is requested from trusted authority
10: Secret key (Sk) ← cpabe.keygen(public key (pk), attr list)
11: Decrypt cipher text using the secret key to get prescription content
12: if ρ.evaluate(attr list) then
13: Pcontent ← cpabe.decrypt(Sk, CT)
14: else
15: Cannot decrypt prescription content
16: end if
17: end for

Prescription Retrieval Steps in FortiRx

- Pharmacy or physician sends different status updates.
- Depending on the type of update, the smart contract is called with the PID as a parameter.
- If the prescription is filled:
  - The smart contract marks the prescription as filled.
- If the prescription needs re-filling:
  - The smart contract requests a refill.
- Otherwise:
  - The smart contract issues a refill.

Algorithm 3 Status Updates for Prescription on Blockchain.

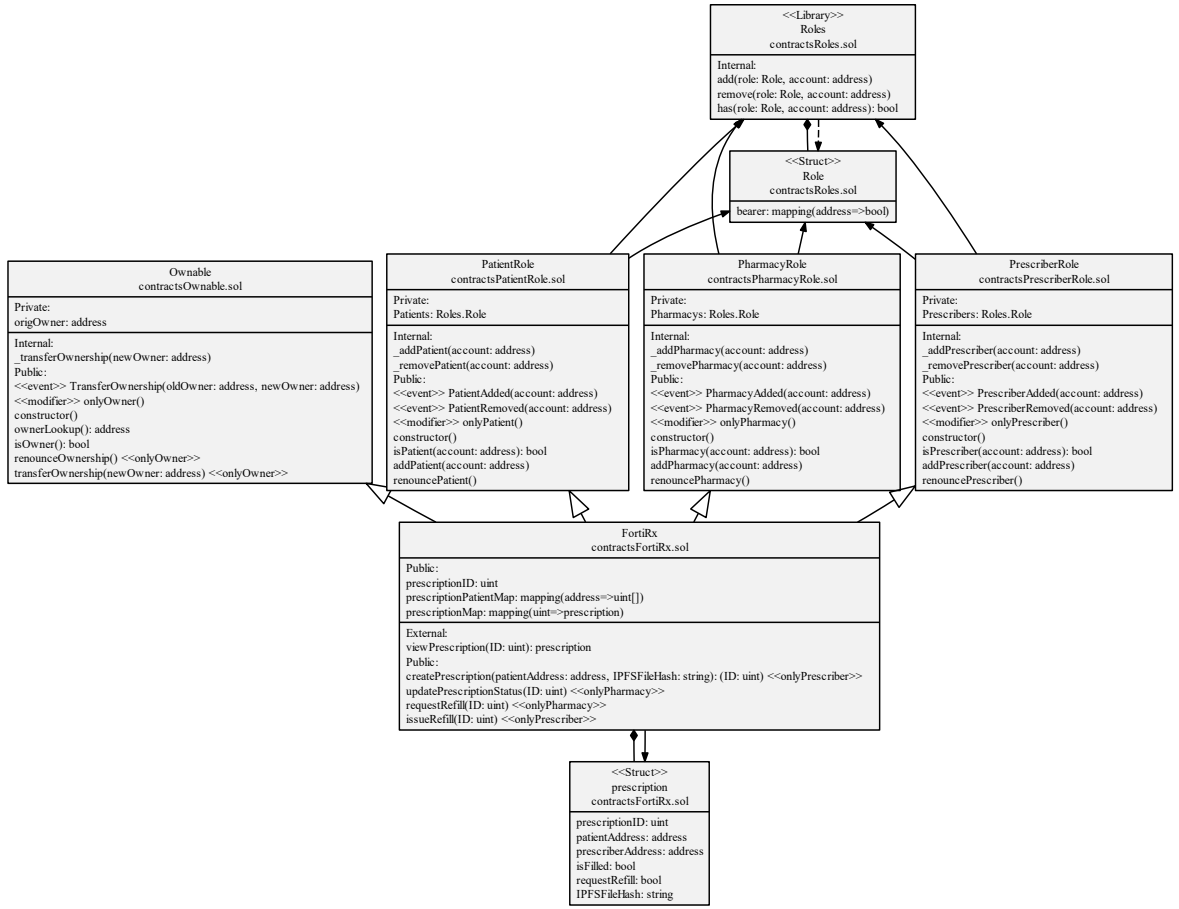
Input: Prescription ID (PID) generated while creating a new prescription in blockchain
Output: The Status of the prescription will be updated

1: Different status flag updates will be sent either by the pharmacy or physician
2: Based on the type of status update, different functions of the smart contract will be invoked with (PID) as parameter
3: if the Prescription is filled then
4: prescription.updatePrescriptionStatus(PID)
5: Smart contract check the pharmacy Ethereum address for access and updates isFilled flag of prescription
6: else if Prescription needs re-filling, then
7: prescription.requestRefill(PID)
8: Smart contract checks the pharmacy Ethereum address for access and updates the requestRefill flag of prescription
9: else
10: prescription.issueRefill(PID)
11: Smart contract checks the physician's Ethereum address for access and updates the isFilled and requestRefill flags of prescription
12: end if

Implementation

Implementation

- Smart Contract Design
  - Solidity language is used for designing smart contracts
  - RBAC mechanism implemented using smart contracts and modifiers
  - Important functions in smart contracts include creating prescriptions, updating the prescription status, requesting refills, and issuing refill



Implementation

- Ethereum platform leveraging smart contracts is used in designing FortiRx.
- Truffle framework to design a Decentralized Application (DApp).
- Local Ganache blockchain used for deploying and testing PoC.
- Performance analysis is performed in public test network sepolia.
- Inter-planetary File System (IPFS) used for off-chain distributed data storage.
- Charm framework is used for prototyping CP-ABE.

Encrypting and Uploading Prescription to IPFS

- A sample prescription Document is created
- The created document is then encrypted using the CP-ABE scheme with a pre-defined access policy.
- The file is then uploaded to IPFS using infra.
- The retrieval process uses Content ID (CID) to retrieve from IPFS.
- Fetched encrypted data is then decrypted.

Results and Analysis

Used Sample Prescription Data

John Doe's Bags of Medications
(Note: you would only know what these are if you accessed an electronic pill identifier site likeDrugs.com)
Morning Ziplock:

- Allopurinol 2 50 mg tablets: learn he takes 1 or 2 a day depending on whether he has gout
- Aspirin ½ tablet: doctor told him to take ½ tablet
- Clopidogrel 75 mg tablet
- Colchicine 0.6 mg tablet
- Glyburide 1.25 mg tablet
- Toprol XL 50 mg tablet
- Amiloride 5 mg tablet
- Enalapril 20 mg tablet
- Tylenol Arthritis 2 650 mg tablets

Afternoon Ziplock:

- Tylenol Arthritis 2 650 mg tablets

PM Ziplock:

- Colchicine 0.6 mg tablet
- Glyburide 1.25 mg tablet
- Simvastatin 80 mg tablet
- Warfarin 5 mg tablet
- Amiloride 5 mg tablet
- Enalapril 20 mg tablet
- Tylenol Arthritis 2 650 mg tablets

Also has:

- Nitroglycerin bottle of 0.4 mg tablets – takes 1 QD or QOD
- Albuterol inhaler: prn. Does not use often.

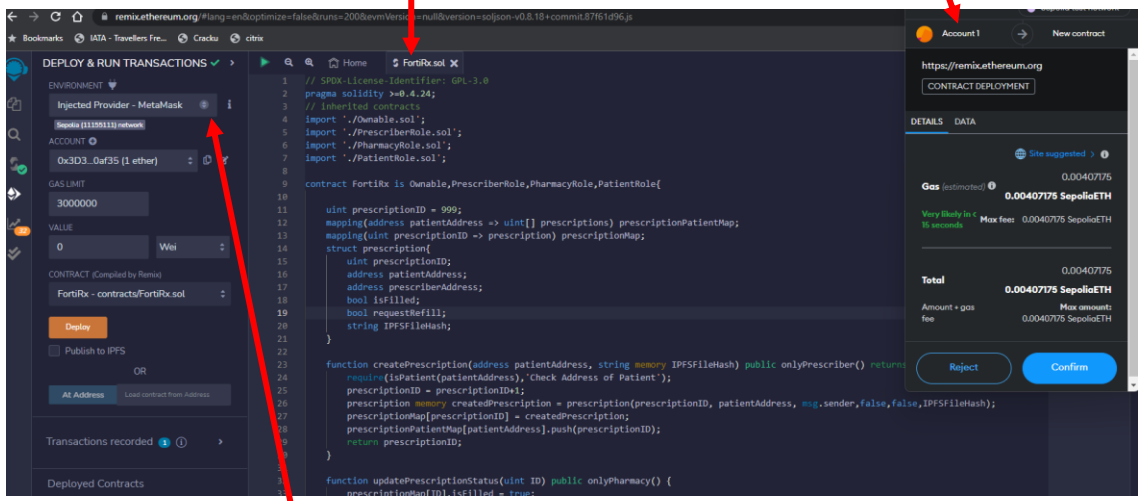
Used Sample Prescription Text
File Size: 913 bytes

Source: https://www.hospitalmedicine.org/globalassets/clinical-topics/medication-reconciliation/1_john-doe-case-prework-for-pharmacist-trx-1.pdf

Encrypting and Uploading Prescription to IPFS

Smart Contract Deployment

Deployment in Sepolia



Ethereum Addresses with Roles

Feature	Value
Physician Account Address	0x3d352313f4f5561d0ffbfda205b52a3c3b70af35
Pharmacy Account Address	0x3D352313F4f5561D0fFBfda205B52A3c3b70af35
Patient Account Address	0x2a9884dfa7E6890FE8AA99FE2486c613C32b697a
Contract Deployment Hash	0x798d1f5ff49f9df09b9856db2646cebc2029d5cd2a45c5ef0c1b9acb9f217c6f
Prescription Content ID	Qme7Sq8gLmE875kE79QyWWFy9wqQ4yHnTEHMur511PrZfF
Prescription Creation Hash	0xda5bd0ce943325696e91bfe140bd8cdd60eafdca6f2a41b07221e499bfe7f1f7

Metrics

- Transaction Time
  - Times taken for a blockchain transaction to be added to a block and safely confirmed
  - Let T_B be the average block time and N is the number of confirmations, Transaction time is computed as:

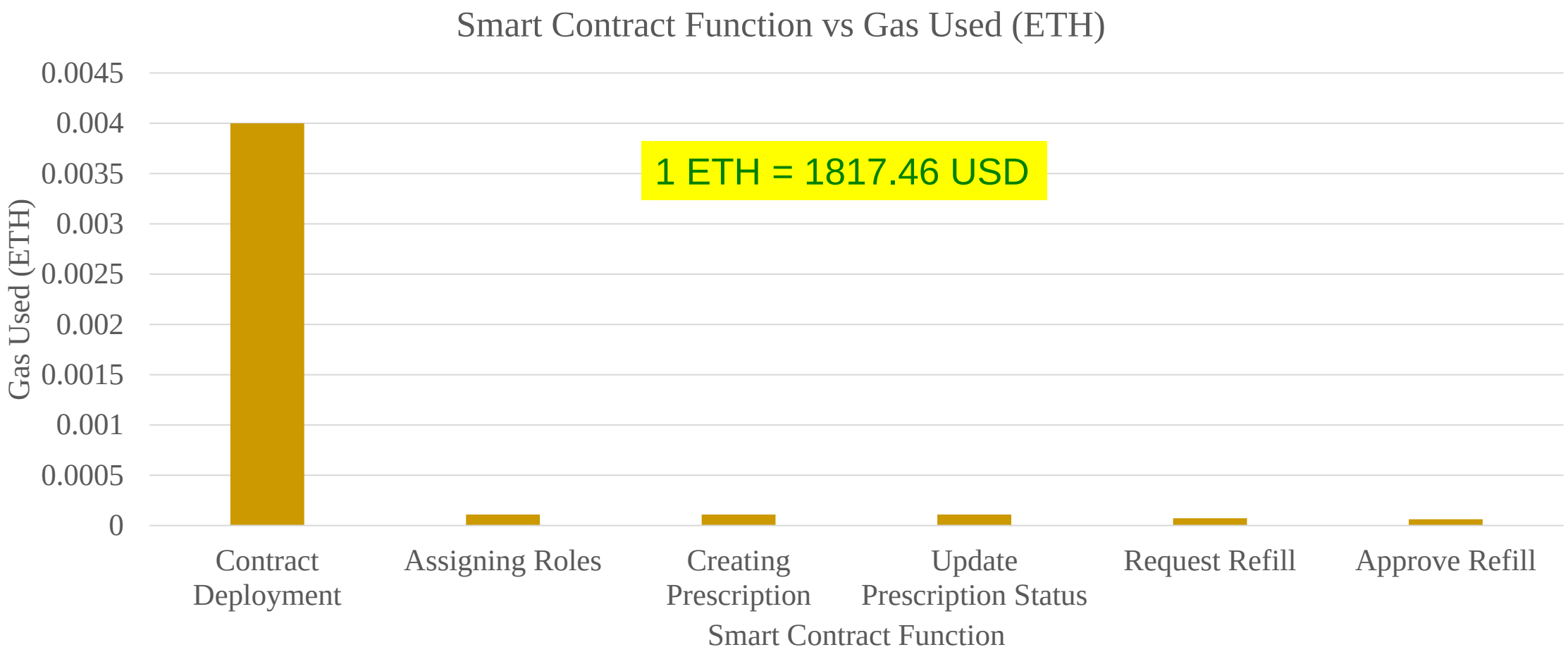
$$Transaction\ Time\ (T) = T_B * N$$

- Gas Cost
  - Represents the computational and storage cost required to execute a transaction
  - Measured in gwei (10^{-18} ETH)

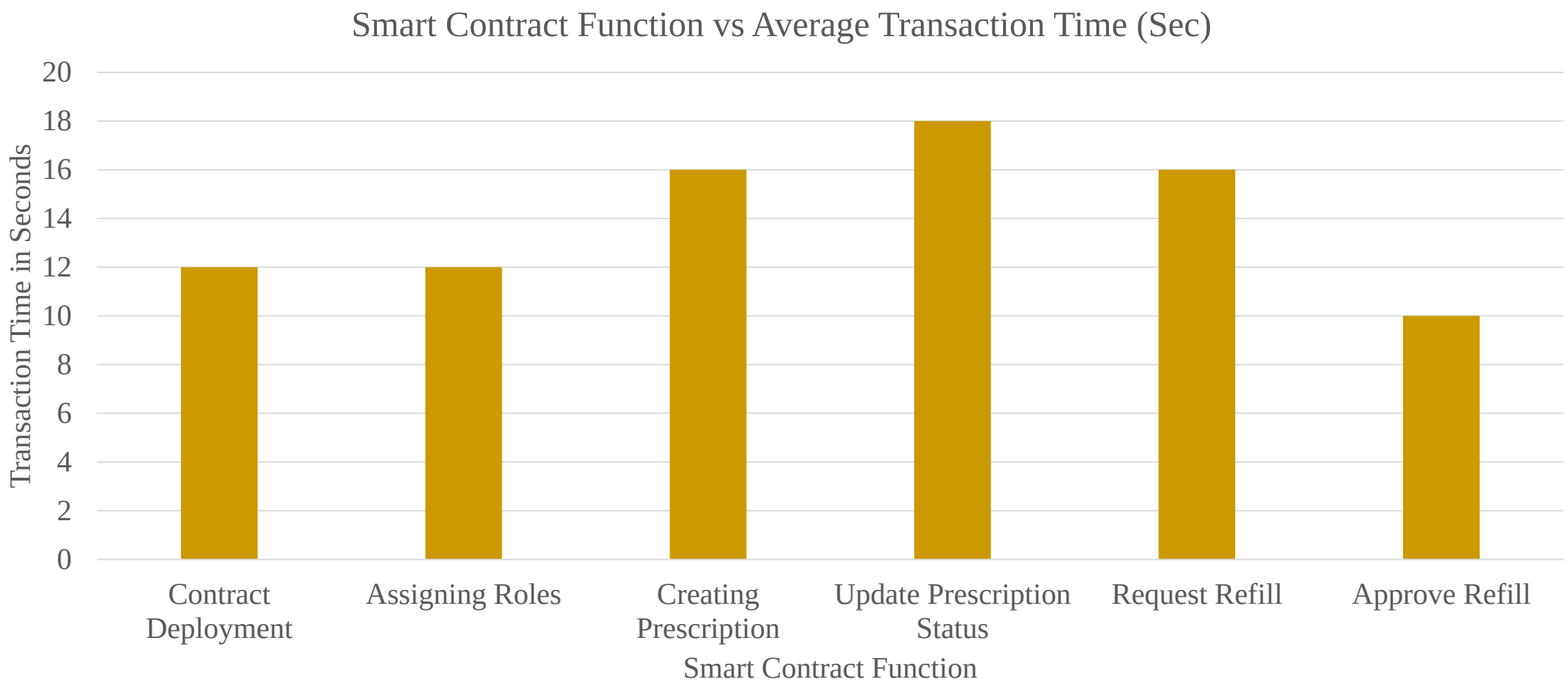
$$Total\ Gas\ Cost\ (gwei) = Gas\ Consumption\ (in\ Gas\ Units) * Gas\ Price\ (in\ gwei\ per\ gas\ unit)$$

Source: Ethereum docs, https://ethereum.org/en/developers/docs/

Transaction Cost



Transaction Confirmation Times



Summary

- Proposed a novel Blockchain-based E-prescription system with smart contract automation of processes.
- Cost overhead of managing large data files is addressed by leveraging off-chain distributed data storage.
- A robust access control mechanism is implemented using Cipher Text Policy Based Encryption (CP-ABE) which allows data sharing between a dynamic group of users.
- Proof-of-concept (PoC) is implemented and analyzed for scalability and reliability in real-world scenarios.

Future Work

- User-friendly GUI will be deployed for easier access to functions
- Decentralized trustless key distribution mechanism should be deployed for efficient attribute sharing
- Analyzing prescription real-time information for efficient demand forecasting models (FortiRx 2.0)

Thank You !!