
FortiRx: Distributed Ledger based Verifiable and Trustworthy Electronic Prescription Sharing

Presenter: Anand Kumar Bapatla

Anand Kumar Bapatla¹, S. P. Mohanty², E. Kougianos³
University of North Texas, Denton, TX, USA.^{1,2,3}

Email: ab0841@unt.edu¹, saraju.mohanty@unt.edu², elias.kougianos@unt.edu³

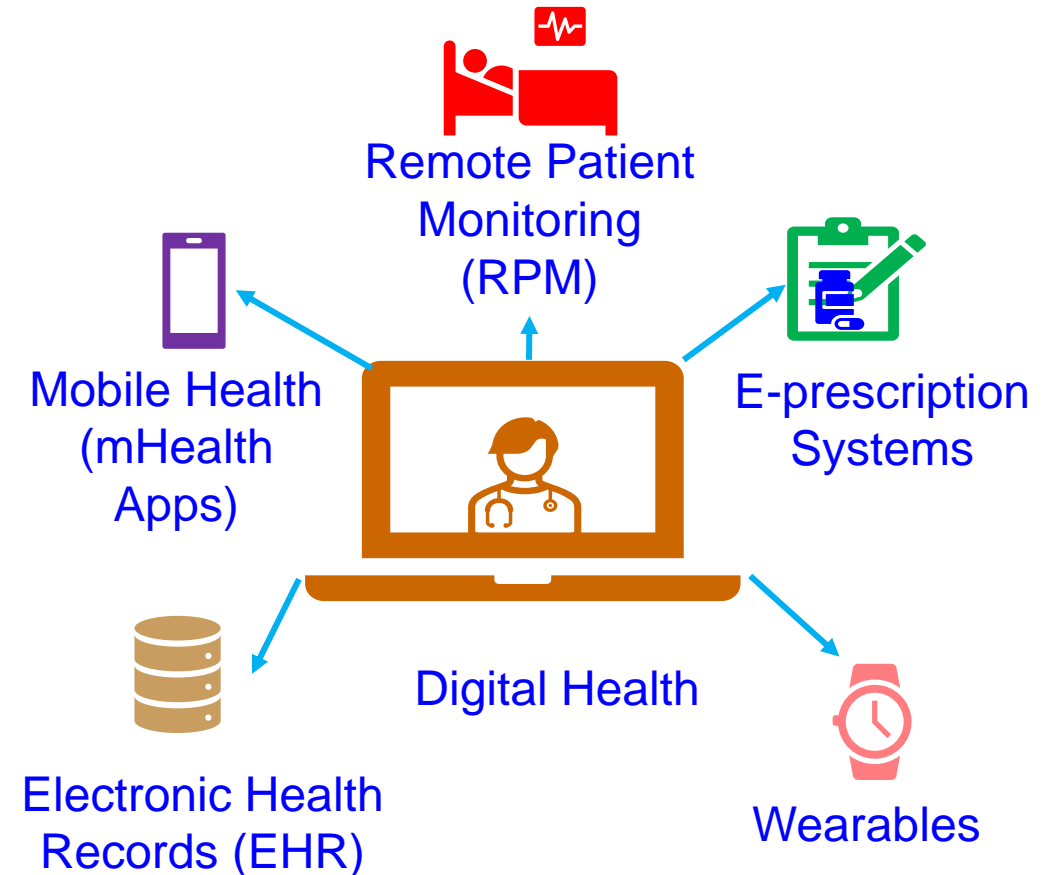
Outline

- 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

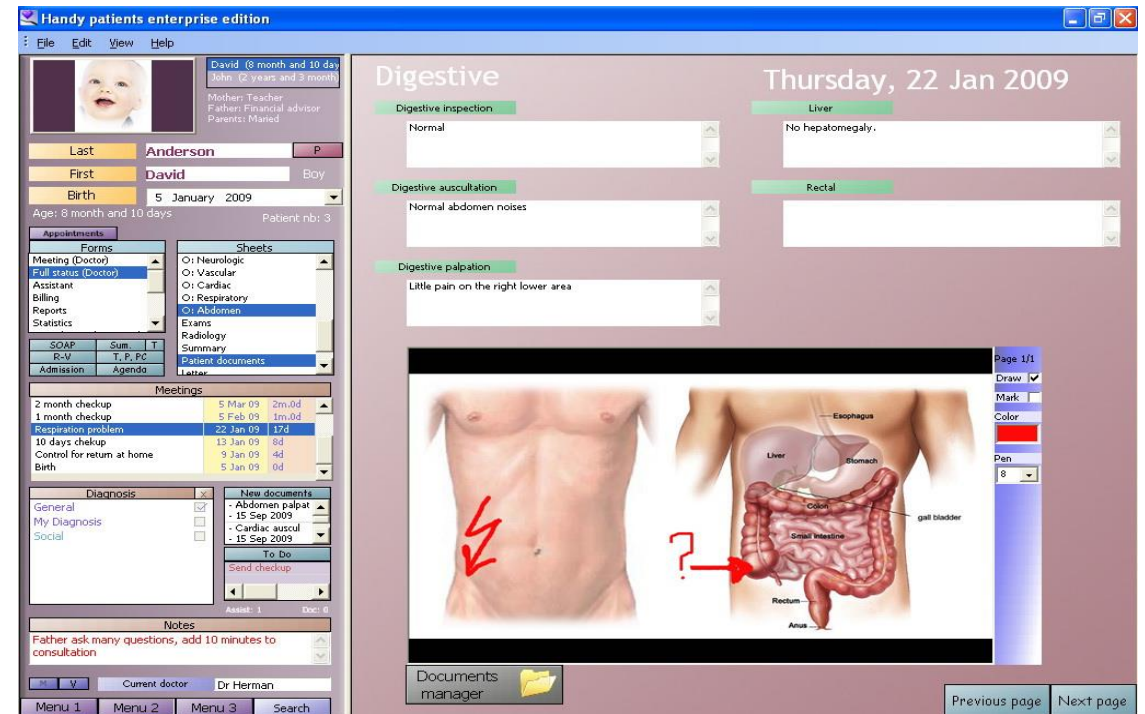
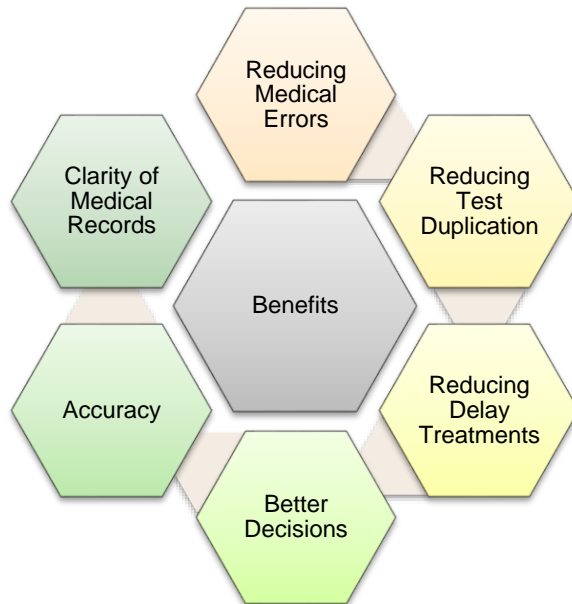
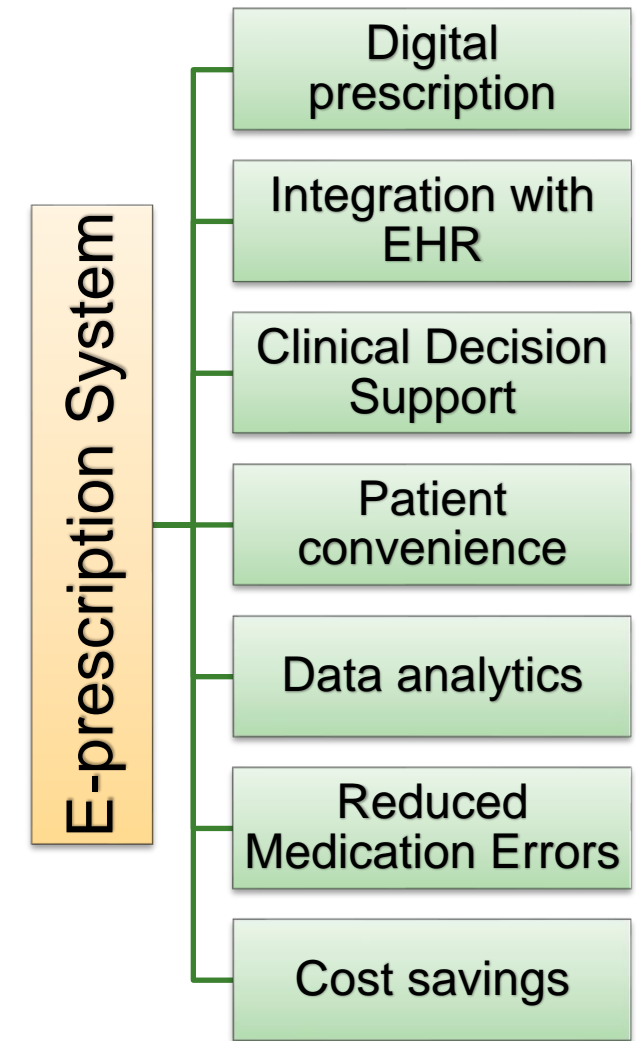


Image Source: DaCarpenter, An electronic medical record example, Handy patients electronic medical record (free open-source version)

Electronic Prescription

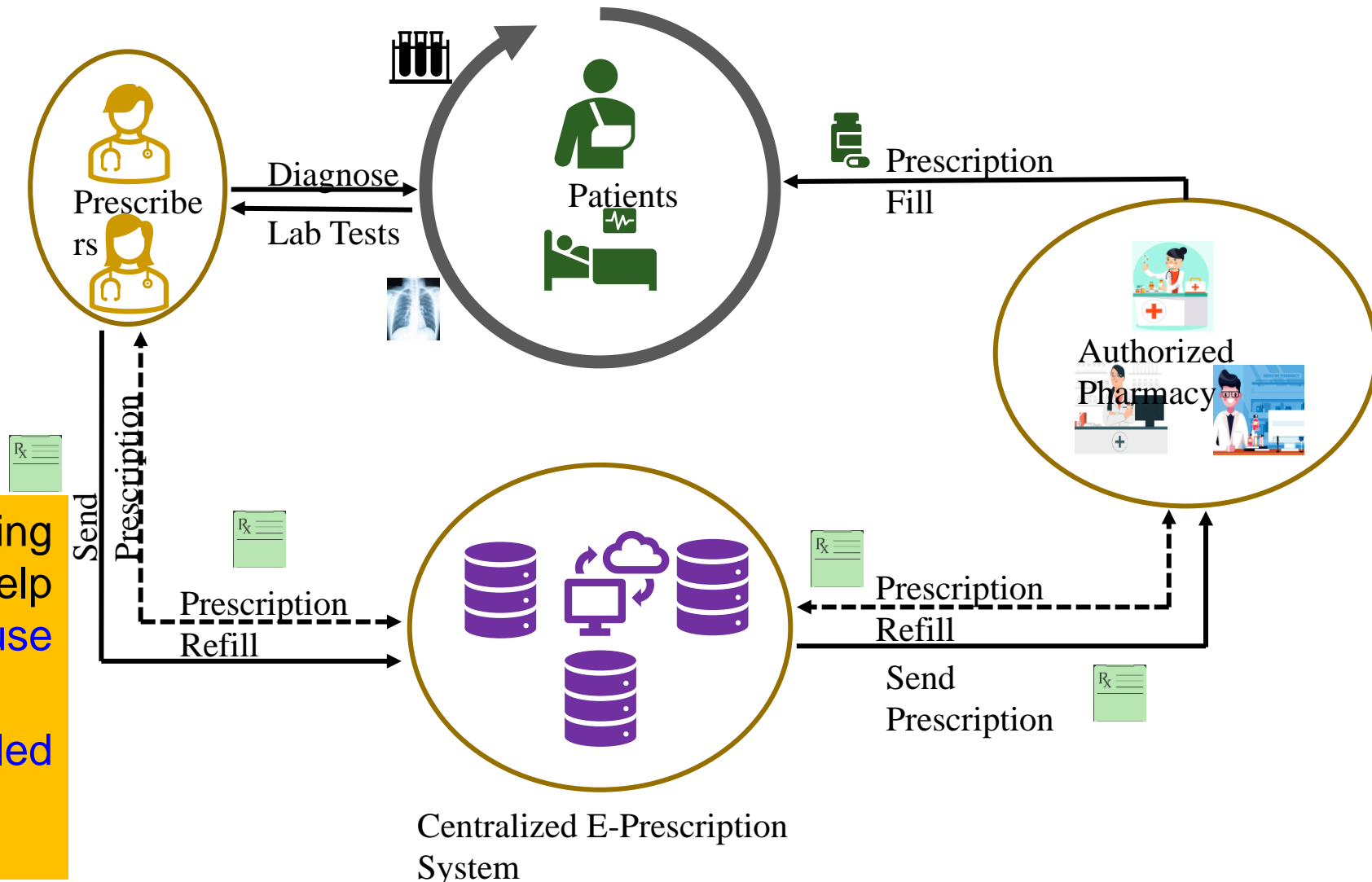
- Revolutionized the way medications are prescribed, processed, and dispensed
- Digital version of prescriptions increase legibility and reduces medication errors
- Clinical Decision Support Tools – Warn potential drug interactions, suggest alternate medication, offer dosage recommendations

- 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

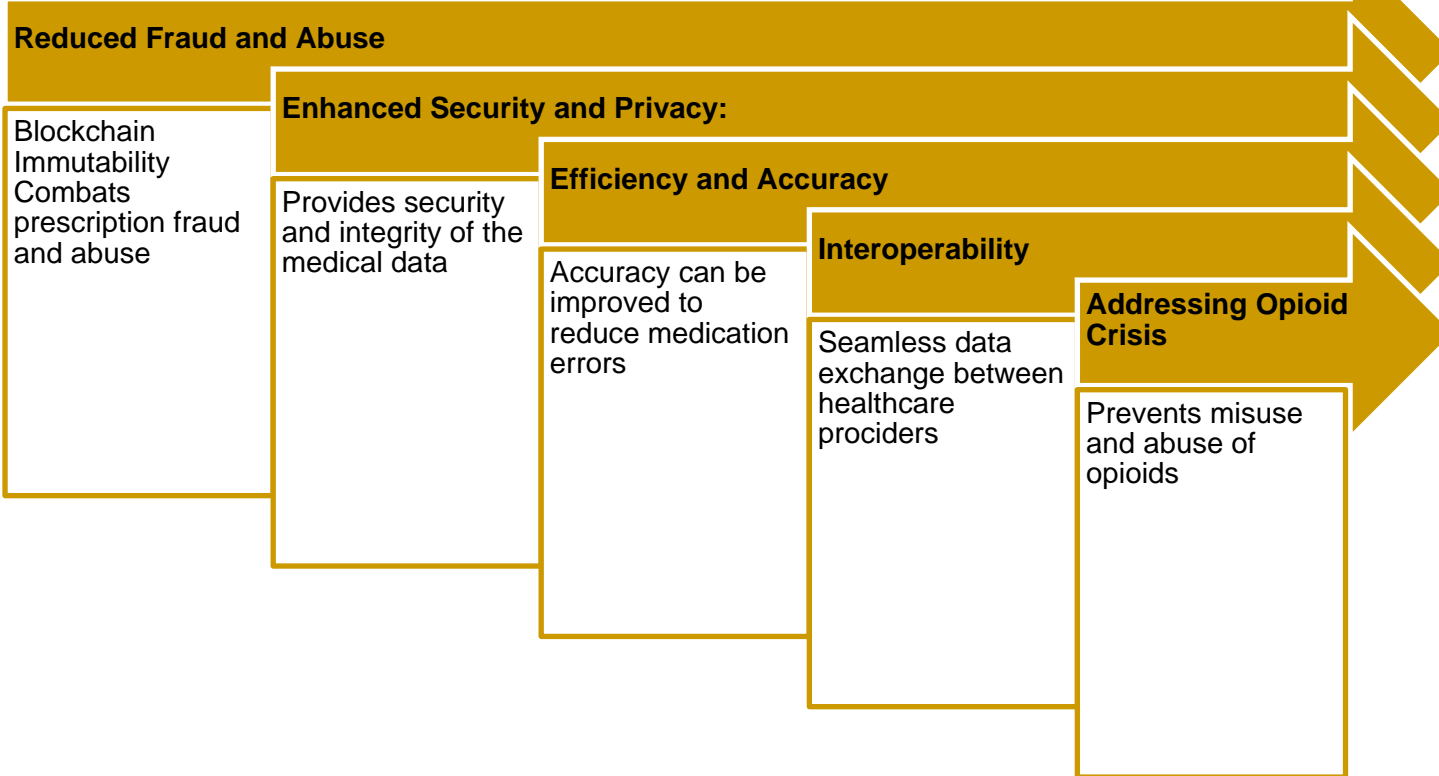


- Prescription Drug Monitoring Programs (PDMP) help mitigate prescription misuse and diversion
- Oversight of controlled substance prescriptions

Motivation

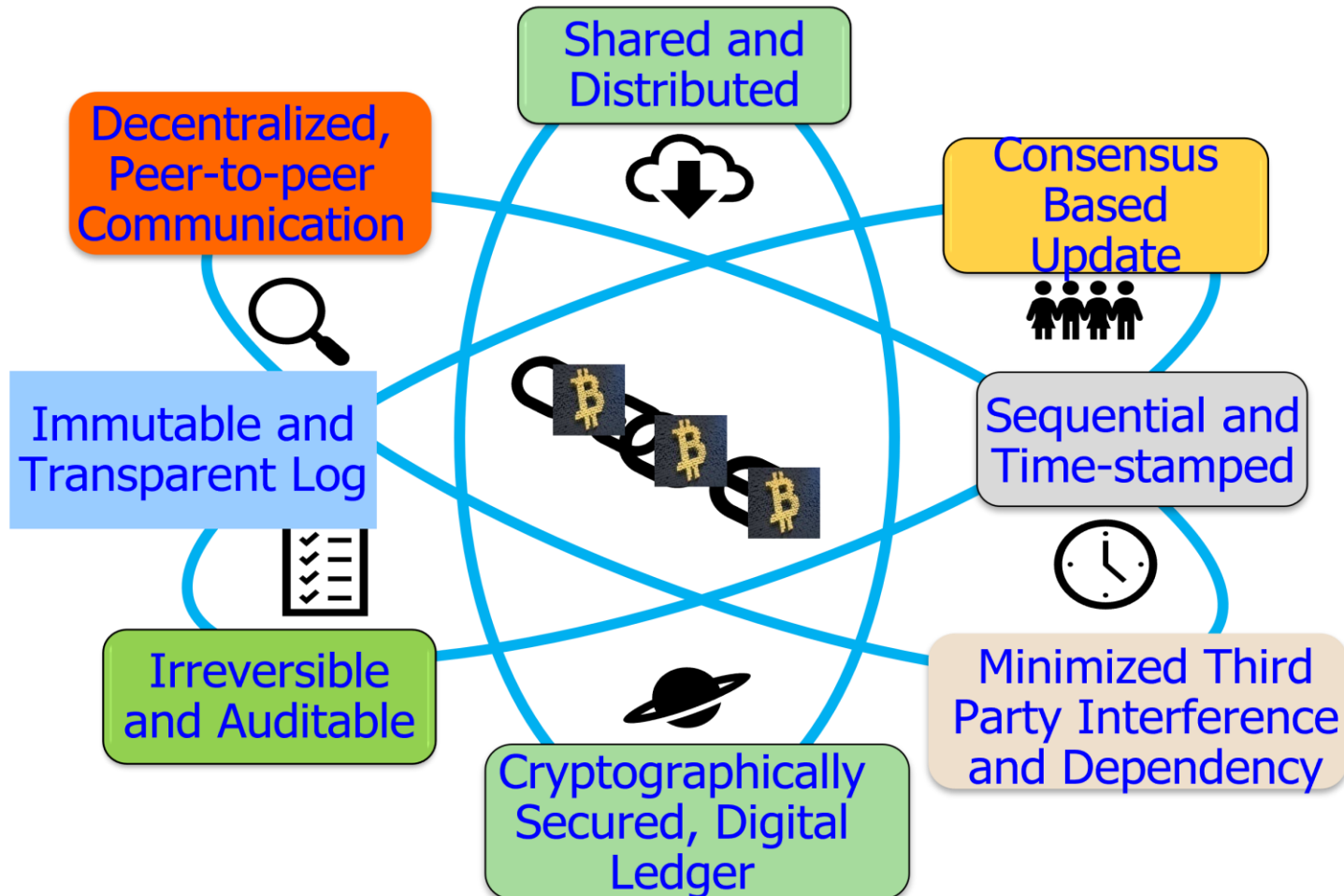
Prescription Drug Type	Annual Abusers	% Among Rx Abusers	% Among Americans
Painkillers	9.7 million	59.5%	3.43%
Opioids Alone	9.3 million	57.1%	3.29%
Sedatives	5.9 million	36.2%	2.08%
Stimulants	4.9 million	30.1%	1.73%
Benzodiazepine Alone	4.8 million	29.4%	1.70%
All Prescription Drugs	16.3 million	100%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

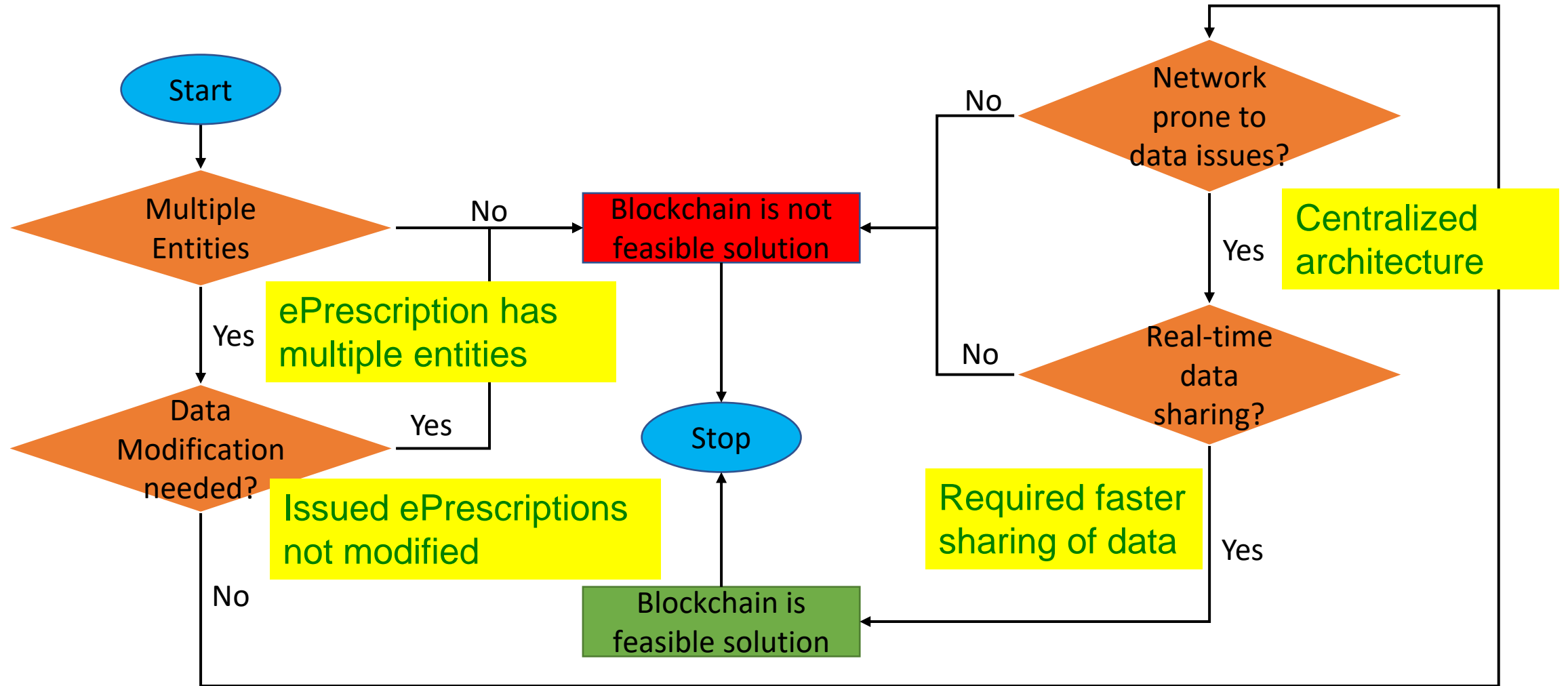
Financial – Accounting Definition: A blockchain is a public, distributed ledger of peer-to-peer transactions.

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

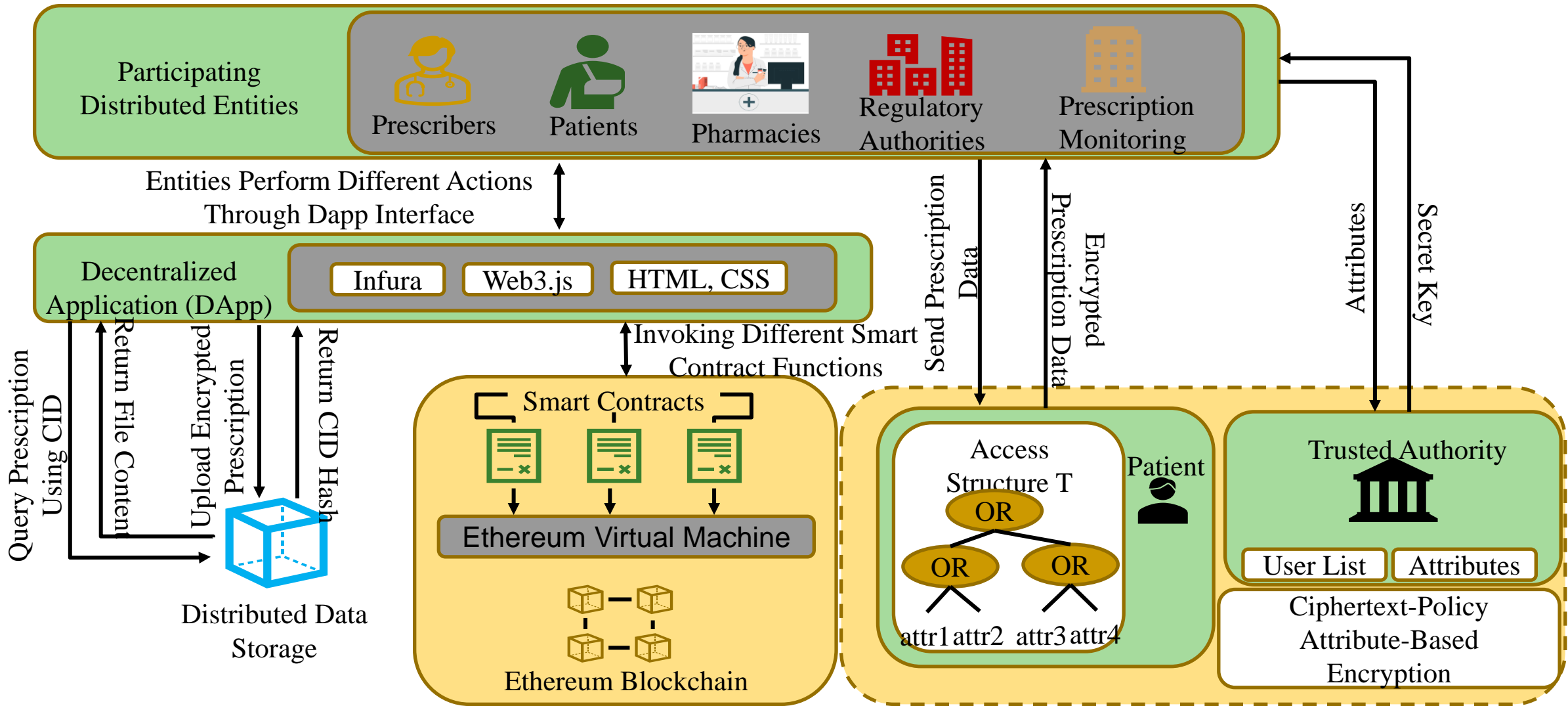
- Proposed FortiRx makes use of blockchain combined with the distributed file system (IPFS) to create a **decentralized environment** for all the participating entities.
- Blockchain enhances the **interoperability** of the system.
- Usage of **off-chain distributed file-sharing systems** to store prescription 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

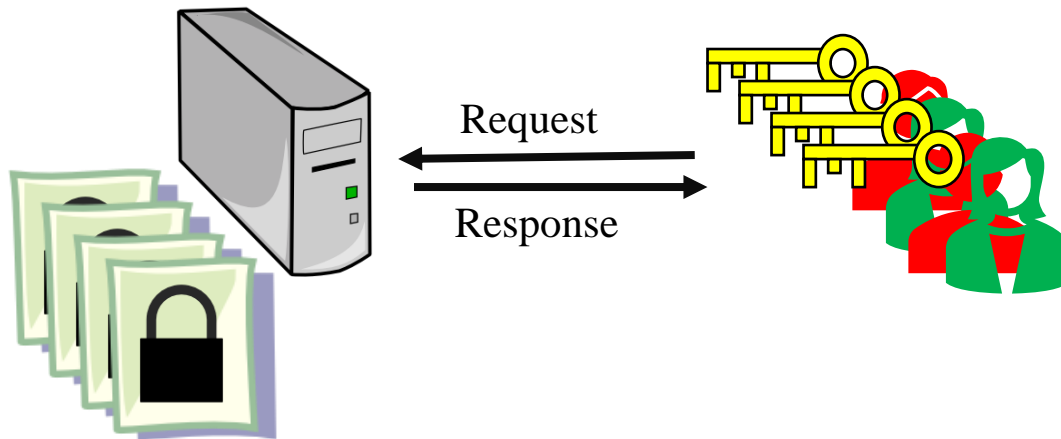
	Blockchain Platform	Smart Contracts	Off-chain storage	Data Privacy	Access Control Mechanism	CP-ABE
Thatcher, et al. 2018 [20]	Ethereum	✓	✗	✗	✗	✗
Musamih, et al. 2021 [14]	Ethereum	✓	✓	✗	✓	✗
Taylor, et al. 2022 [19]	Ethereum	✓	✗	✓	✓	✗
Aluaimi, et al. 2022 [3]	Ethereum	✓	✓	✗	✓	✗
Ionescu, et al. 2022 [11]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

Architecture

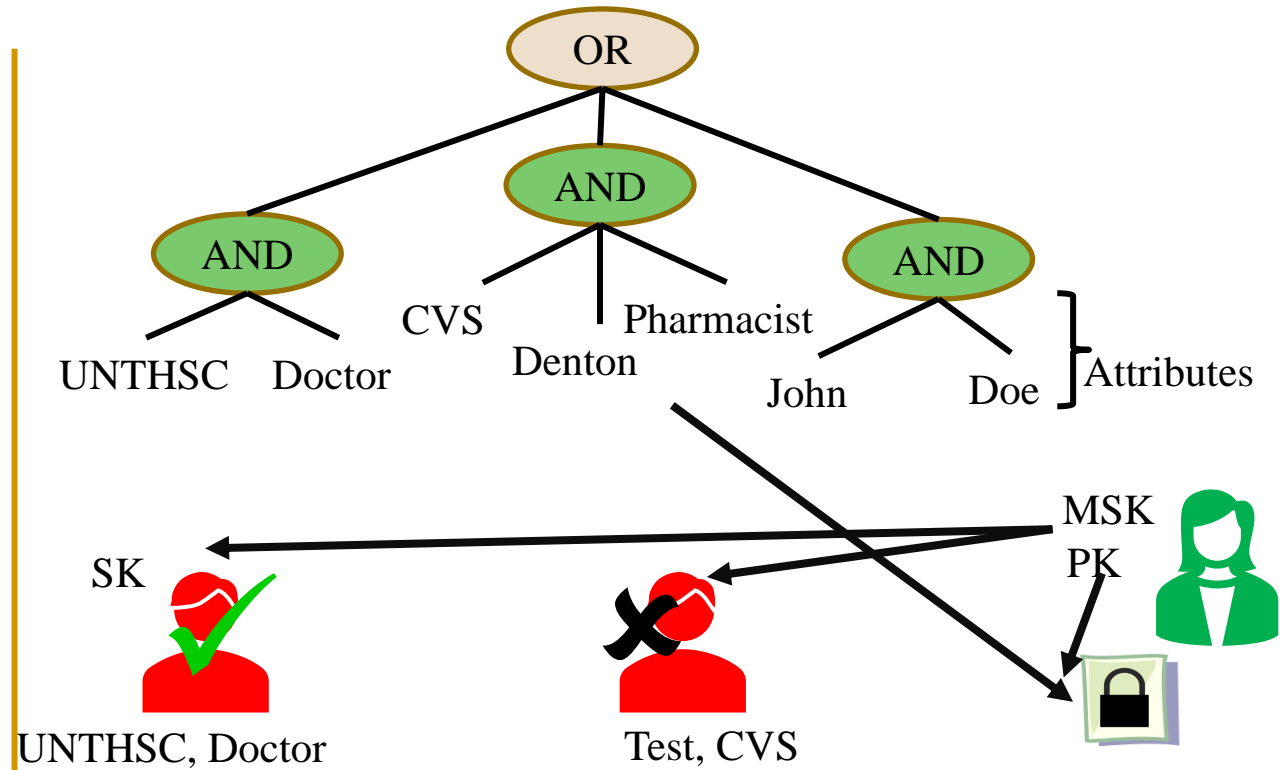
FortiRx Architecture



Asymmetric Encryption vs CP-ABE



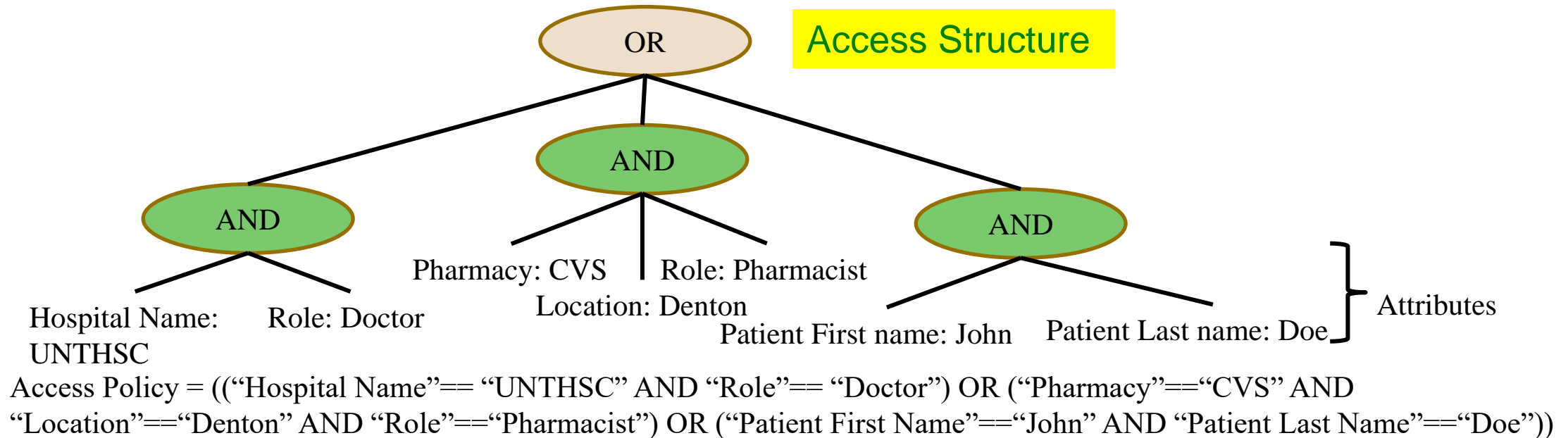
- Secure but not flexible
- New key for every participant
- Fine-grain access control not possible
- Needs efficient key distribution



- 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

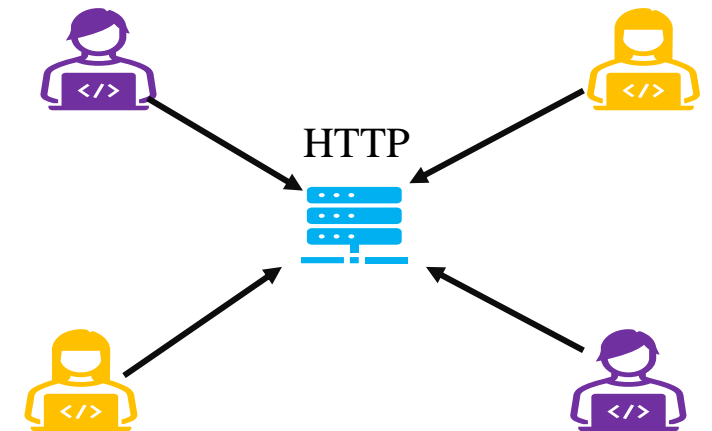
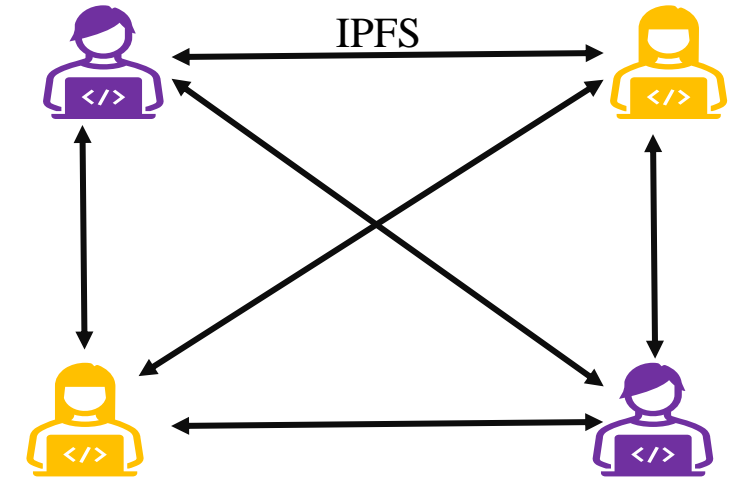


Access Control Mechanisms

- 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 p defined by the patient

Output: Content ID for IPFS file, Transaction hash of prescription creation in blockchain

```
1: A digital prescription is generated, and a file is created
2: For each prescription file f do
3:     Open file in read mode
4:     FileItem  $\leftarrow$  open(filePath,'r')
5:     Read prescription content from the file
6:     prescription content (Pcontent)  $\leftarrow$  fileItem.read()
7:     Encryption is done using the public key (pk) and policy  $p$  to generate ciphertext of the prescription content
8:     Cipher text CT  $\leftarrow$  cpabe.encrypt(pk, Pcontent , $p$ )
9:     A new file is created and generated cipher text is written to that file
10: end for
11: For each encrypted prescription file f do
12:     Send upload request to IPFS
13:     Response (res)  $\leftarrow$  requests.post(Infura endpoint, authentication parameters, file f)
14:     Content ID from response is retrieved
15:     Content ID (CID)  $\leftarrow$  res.text['Hash']
16: end for
17: Prescriber creates a new createPrescription transaction in prescription smart contract
18: Transaction (Tx)  $\leftarrow$  prescription.createPrescription(patient address (Paddr),CID)
19: if caller == Prescriber then
20:     A new prescription is created and added to patient's address
21:     Emit an event (ev) with prescription data and a log is generated
22:     Return transaction hash (Tx hash)
23: else
24:     Reject Tx
25: 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  $\leftarrow$  prescription.viewPrescription(PID)
4:     Get IPFS Hash (CID) from the function response
5:     CID  $\leftarrow$  Pret['IPFSHash']
6:     Send a request to IPFS to retrieve prescription content (Pcontent)
7:     Response (res)  $\leftarrow$  requests.post(Infura end point, CID, authentication parameters)
8:     Retrieved cipher text (CT)  $\leftarrow$  res.text
9:     Secret key for a set of attributes attr list is requested from trusted authority
10:    Secret key (Sk)  $\leftarrow$  cpabe.keygen(public key (pk), attr list)
11:    Decrypt cipher text using the secret key to get prescription content
12:    if  $\rho$ .evaluate(attr list) then
13:        Pcontent  $\leftarrow$  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

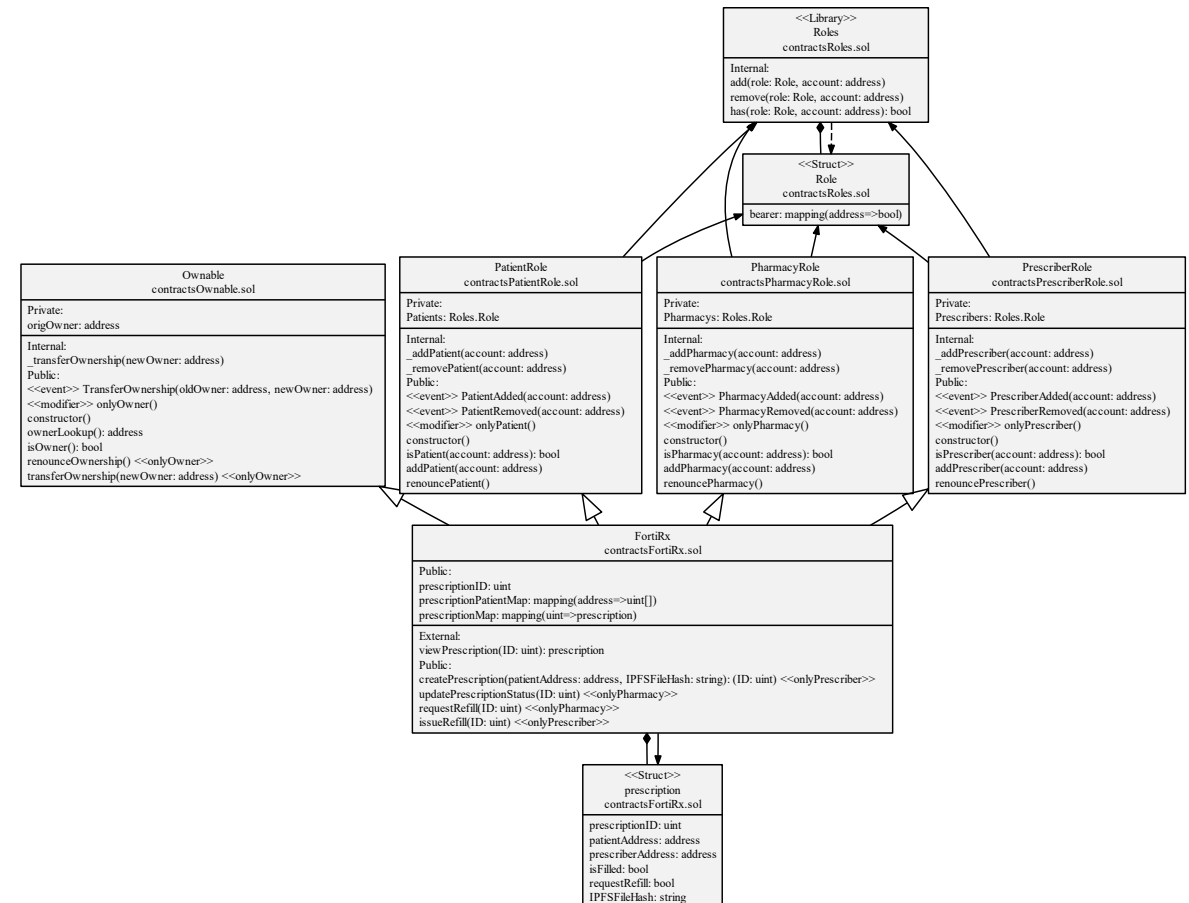
Output: The Status of the prescription will be updated

```
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 like Drugs.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

The screenshot displays a terminal window with the following content:

```
Encrypted Data  
[ 'c1': [ 'C_tilde': [32156964776025433085096874722998427334474690479036758738867865391736110265035806644773102421649340468956817059090697756452955137114836886501609304445748, 8375664003792690789830315319691246205565349842994900409338517553806195602057786851021342716938948436393201837227569793818305191624227910299357750124601], 'C': [3545286006617797074529535372798534778334137782844731531763005399874507628734915994424113705903400007883082931881184216803491065802701878551738836198921870, 835705397127054880189874280145155803809908146900262686846758074193556406824968134973723098585327392013211507111859764557659959088159491163104986261305214], 'Cy': { 'ONE': [421717988532050626331214696361518319151210055320261156845565025340893484499515370690704516347679968828775679626051286668260953499204993563587563283619, 521411298979699148156064926499203539413429517230529788341220697700263863661588679408536998667176963139130467454735515028415268994174370938814309913794843], 'THREE': [421717988532050626331214696361518319151210055320261156845565025340893484499515370690704516347679968828775679626051286668260953499204993563587563283619, 521411298979699148156064926499203539413429517230529788341220697700263863661588679408536998667176963139130467454735515028415268994174370938814309913794843], 'TWO': [776248738852230528423976894319937197482496554704928576617100447522635514899703444542313563612682920869926404282215546936496015506735462602279385429747926, 811814252755172149204979014182244586580469082343534043013471943922402107540724704693686375820755678411209976200944280682568090336860288528588108636765908], 'FOUR': [776248738852230528423976894319937197482496554704928576617100447522635514899703444542313563612682920869926404282215546936496015506735462602279385429747926, 811814252755172149204979014182244586580469082343534043013471943922402107540724704693686375820755678411209976200944280682568090336860288528588108636765908], 'Cyp': { 'ONE': [3462661131960550795121839248675829474122878610292834400114018667171825605253080081037893556941071298311923283299152606716901449101714563002436022383875549, 3844230834399246056765123644963390648806844311928259016727500969957584191314230525161099302872720438135507616836089460582888374449758314100827252215751090], 'THREE': [75778580573190155391122791396145525255657871960487735726477720490342917804349696138452610459546563858960266134745882515819503507380778065462926187730658403, 10801058640177912383430790832492492313938988247724311998991797113424481055087320699428305724449140293543089443701536449039254610444906603481227280721629], 'TWO': [8344776533390479676082066171109501642234510879127557540865833528688235358015924755773886449395718065820142364107517989624398404789041439504609573675701538, 8607236712561655080573298955367766900305949151015256622378543278012801337214633184234746936356553254032732562497913359353608713961993343064197039244719], 'FOUR': [8064280825192130672032932068400289718329034108824984457467435311752920179567656372544882832421463582743642008109993448545751611723264241486021136894681073, 3561827273631685939592151147010324151126824743991426854170279092465456120594687755108122993849588458094019627359816582456598437223146568156560112031271351], 'policy': ['(ONE or THREE) and (TWO or FOUR)', 'attributes': ['ONE', 'THREE', 'TWO', 'FOUR'], 'c2': {'alg': 'HMAC_SHA2', 'msg': {'"ALG": 0, "MODE": 2, "IV": "3xchsyuYidXRtKE/wdWcXg=", "CipherText": "1dKUF1i/IAQ G0lqe16ZnVxBJP+25Fuy3YfZcFjBBys2GdnvSQV6s3M4ofIVGIBI9790UNRRzYsRL6FoAlIIVs3XdmJzCjvS2tXhVgZB/Ux789XfTPkz60K7eUl4q8Fv0r2ChSi1EaTXiFID0tEmMAjY1zjL58yVxm3G/zsYDqoaevtaWqtu7YbZ5p060be1qIA0YC1DHL F4vTIoPqXm6nqzSYNNWw4R6+Ui+5DA0kIrClvLjLLySziVsoo+nC12LSIAIATvgY40CMKYomfKvXrECAP90832f0c7bqTtpvBN0RQbdAyHGoVes6fsLaeX1LjLDcn/vWtqg7+ls+Sto8M0oo0tlvG2qjsot0nll7VbvtkmrnxwllVsq2xtenx8P6RE55d0QCXFRH 7UMGEPHSt0EVrAXFmWlWlFqctOM9tu4J1zc/CI72P5BRw0rme803Kf10hu3F810vB0etuo6r3zAdCn4tyY2jy/Sl0dLnQarJtCUB2CE-Vz+RRQYPM+PltoYUvuhfGpBeW0gKuvh010yVqrMGVjOKLmFYL4RF+K9UNfINT84msD/UUoGdEKUoU7NeM4ki9s/rZERGdHLKtSnczw0qe+LTXue4AjgdAIYglDx4dVK+TCch/qAychMddj+wIBE0xeh89Xg1cnlqAVA5YJpJnBfFLmP5qwZ5tg+8DtXhLuvjqldFmaEVlqcW82yUaV0Cccbtu+iu16D7m3KGSkovJALZ+i89Hsf0chUuXabP3ysyiA8y8SZ7S8M0P51/ubQ4k0jMEHG VI4m2a6tk3ALuTBcrZZF/hemt2/mSn/wfbsQC4A+Hb9P57fn/GY+P8hdPjFO0+HeYxyN4xrY5EJACJY0qnvbhYQRJaEFLEX5BULNS565Z0MAqha+gzdCMXqJUuYjwzTAKHfrevAwLb157VVOXLDK4FYWJRHKHotWA3gaCnMhBTPYfLUV2upIITCSkn73bX1y11a8CwJsv4lsgHASqjehFlebF0zf1mzc3JXiqzCcZLHvD4hbgVee1h5CRz0QvQnJabGSHWj/57FqX+zCSD3Fus/cMPrEGGjJMSN7HmvyV6icjATQw4d+OzzM73GNMVMcInttFPAIG858wXmNz2YNN1ZVusjVm/rFFafvDnHfFCY10GFswUSGFAVzyTwPKZyXMGWmgzrUaaLNxz0g=""}, 'digest': '20776da162fb1e825c56dad07f1c1c33bc776ad7a6c0c7da14468441e4f95'}}]
```

Content ID
<Response [200]>
Prescription.txt: Qme7Sg8LmE875Ke79QyWfY9wQ4YnTEHMur511PrZFF
Folder CID: QmWP13wr64ft1nT7PUPM3wxBr0s5x1Lv1jzWg1zFNXTJFRH

```
Decrypted Data  
[2055518218368535312257156353032542535393806874053072486268224518005117455169046211829527488705937844597456797852989786590374842683211657473035663777879271, 3720114716169197903951888851439982024564117839553509220866437609836832652740866847294379841501181255853864519743502467547014029491057158033532387391522880]  
b'John Doe\x20\x99s Bags of Medications\n(Note: you would only know what these are if you accessed an electronic pill identifier site like\nDrugs.com)\nMorning Ziplock:\n\n\x20\x80\x2a2 Allopurinol 2 50 mg tablets: learn he takes 1 or 2 a day depending on whether he has gout\n\n\x20\x80\x2a2 Aspirin \x2c\x2b\x2d tablet: doctor told him to take \x2c\x2b\x2d tablet\n\n\x20\x80\x2a2 Clopidogrel 75 mg tablet\n\n\x20\x80\x2a2 Colchicine 0.6 mg tablet\n\n\x20\x80\x2a2 Glyburide 1.25 mg tablet\n\n\x20\x80\x2a2 Toprol XL 50 mg tablet\n\n\x20\x80\x2a2 Amiloride 5 mg tablet\n\n\x20\x80\x2a2 Enalapril 20 mg tablet\n\n\x20\x80\x2a2 Tylenol Arthritis 2 650 mg tablets\n\nAfternoon Ziplock:\n\n\x20\x80\x2a2 Tylenol Arthritis 2 650 mg tablets\n\nPM Ziplock:\n\n\x20\x80\x2a2 Colchicine 0.6 mg tablet\n\n\x20\x80\x2a2 Glyburide 1.25 mg tablet\n\n\x20\x80\x2a2 Warfarin 5 mg tablet\n\n\x20\x80\x2a2 Amiloride 5 mg tablet\n\n\x20\x80\x2a2 Enalapril 20 mg tablet\n\n\x20\x80\x2a2 Tylenol Arthritis 2 650 mg tablets\n\nNALS troglycerin bottle of 0.4 mg tablets \x2c\x20\x80\x93 takes 1 QD or QOD\n\n\x20\x80\x2a2 Albuterol inhaler: prn. Does not use often.\n'\n\nosboxes@osboxes:~/desktop/FortiRx$
```

Encrypted Prescription

Content ID from IPFS

Retrieved Prescription Information

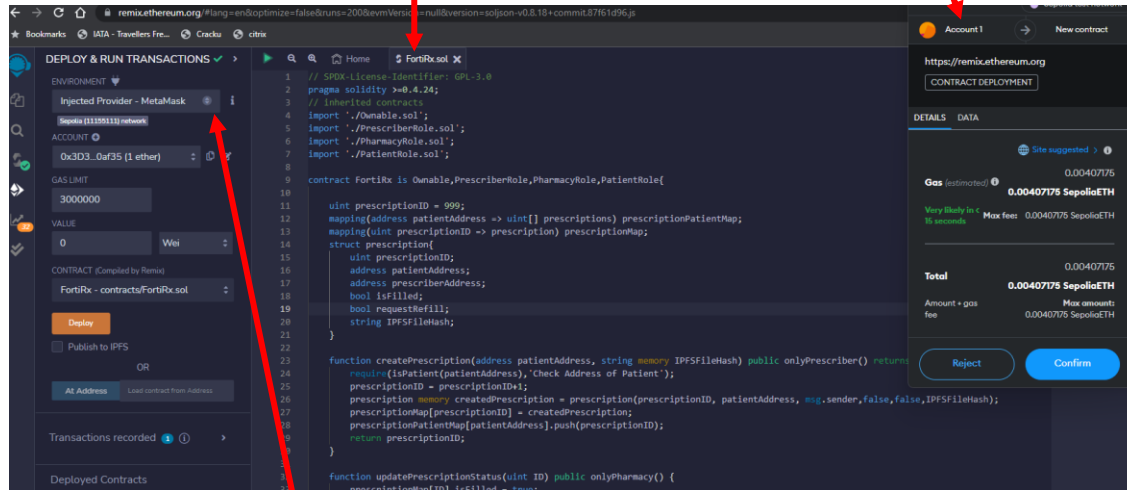


Smart Contract Deployment

Deployment in Sepolia

Smart Contract

Wallet Transaction



Remix Environment
Network Configuration

Ethereum Addresses with Roles

Feature	Value
Physician Account Address	0x3d352313f4f5561d0ffbda205b52a3c3b70af35
Pharmacy Account Address	0x3D352313F4f5561D0fFBda205B52A3c3b70af35
Patient Account Address	0x2a9884dfa7E6890FE8AA99FE2486c613C32b697a
Contract Deployment Hash	0x798d1f5ff49f9df09b9856db2646cebc2029d5cd2a45c5ef0c1b9acb9f217c6f
Prescription Content ID	Qme7Sq8gLmE875kE79QyWWFy9wqQ4yHnTEHMur511PrZfF
Prescription Creation Hash	0xda5bd0ce943325696e91bfe140bd8cdd60eafdc6f2a41b07221e499bfe7f1f7

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:

$$\text{Transaction Time } (T) = T_B * N$$

■ Gas Cost

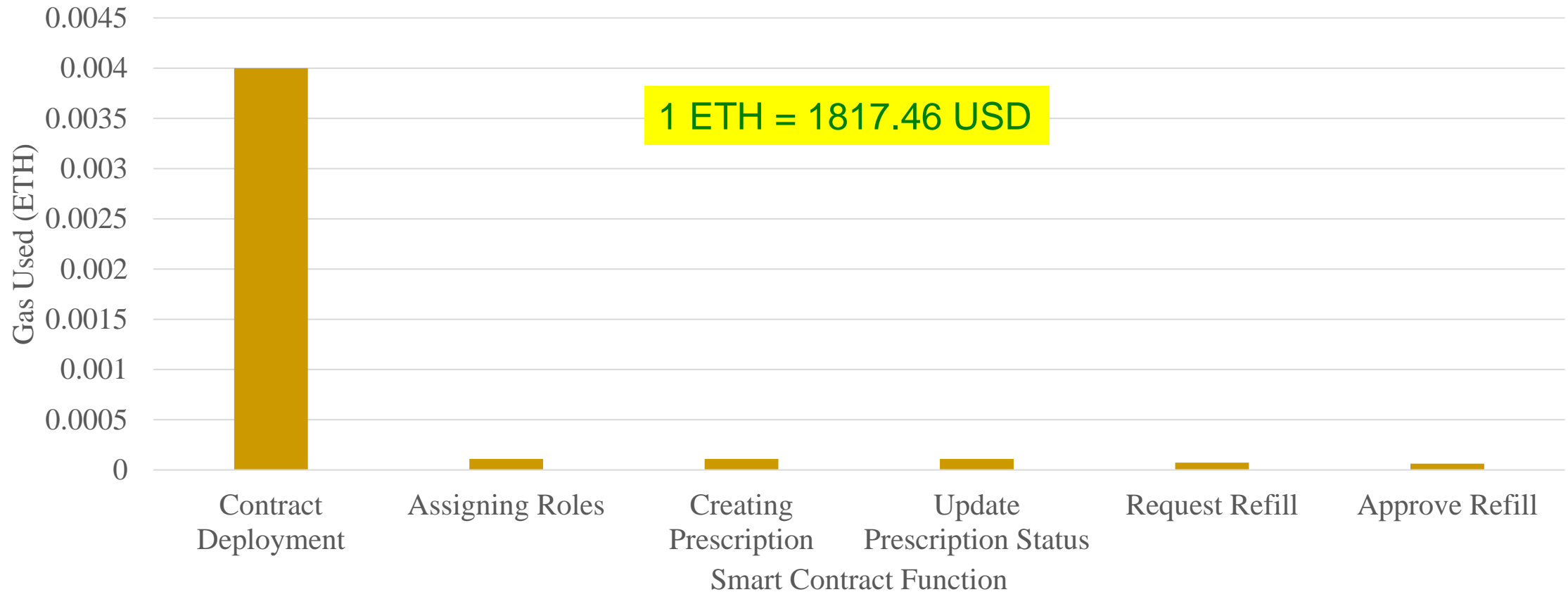
- Represents the computational and storage cost required to execute a transaction
- Measured in gwei (10^{-18} ETH)

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

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

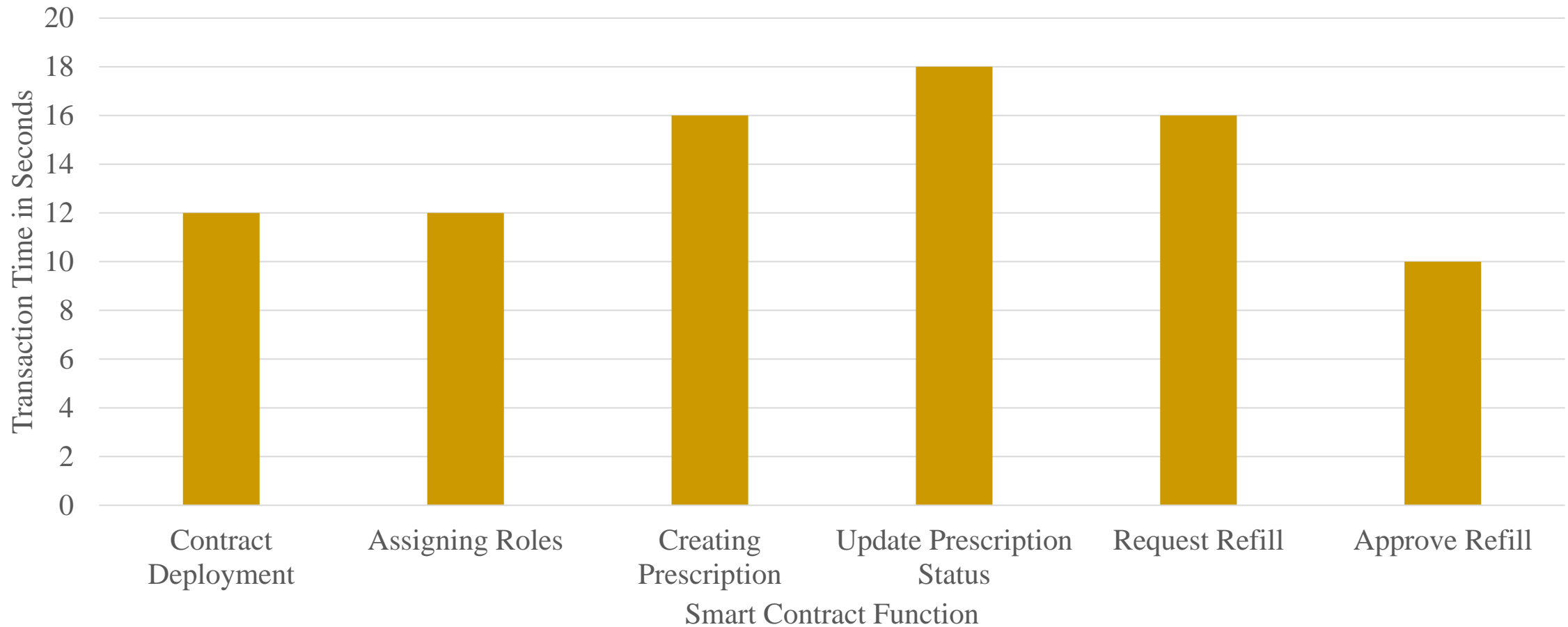
Transaction Cost

Smart Contract Function vs Gas Used (ETH)



Transaction Confirmation Times

Smart Contract Function vs Average Transaction Time (Sec)



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 !!