

Бизнес модели с блокчейн

проф. д-р инж. Венета Алексиева

ОСНОВНИ МОМЕНТИ

- Бизнес приложения с blockchain
- Изисквания за имплементация
- Модели на имплементацията
- Blockchain и IoT



- В този момент базираните на блокчейн решения са на различни етапи на приемане:
 - Изследване
 - Пилотни решения
 - Разработка
 - Производство
- Повече от 30 световни компании имат напълно функционални продукти, изградени на Blockchain

Директиви на ЕС за DLT



- **Енергоспестяващи и съобразени с околната среда приложения** – прилагане на енергоефективен консенсусен механизъм при транзакциите.
- **Транспорт** – логистика, интелигентни застраховки, оптимизация при зареждане на електромобили.
- **Здравеопазване** – ефективността при докладването на клиничните изпитвания, оперативна съвместимост на електронните здравни данни; подобро разпространение на лекарства; контролируемост на лични здравни данни от страна на гражданите
- **Вериги на доставки** - улеснява препращането и мониторинга на произход на стоките, техните съставки или компоненти.
- **Образование** – въвеждане на криптирани образователни сертификати и системи за трансфер на академични кредити
- **Творчески индустрии и авторски права** – проследяване и управление на интелектуалната собственост, улесняване на защитата на авторските права и патентите
- **Финансов сектор** - повишава прозрачността, намалява разходите за сделки, преустановява скриването на печалби

Технологичните предизвикателства

- Липса на зрялост
 - За повишаване на доверието към системата е необходимо разработването на алгоритми за повишаване на стабилността и устойчивостта, особено за транзакции с голям обем
- Мащабируемост
 - В мрежите без разрешение се наблюдава по-ниска мащабируемост, ограничени скорости на проверките и на потока данни
- Латентност
 - необходимо е намиране на механизъм за едновременно обработване на повече транзакции (консенсусен алгоритъм PowerGraph, базиран на DAG с ниска латентност при комуникация с DLT)
- Оперативна съвместимост между различни DLT
 - разрешаване на транзакции между контрагенти по равнопоставен начин и валидиране на идентичност;
 - осигуряване на еквивалентност между умните договори и уеднаквяване с реалната правна терминология;
 - интерфейси между множество разпределени книги;
- Сигурност
- Управление

Регулаторни предизвикателства

- Развиваща се нормативна база:
 - Променящи се закони
 - Неясна юрисдикция
 - Регулаторни стандарти
 - Механизми за анулиране
 - Интелектуална собственост (дизайн, бранд, патент, авторство, копиране, протекция...)
 - идентифициране на всички участници - КҮС
- Недоразвит регулаторен контрол
- Развитие на индустриалните стандарти за DLT
- Мерки срещу „пране на пари“

Социални предизвикателства

- Поверителността на данните –лични, медицински досиета
- Правото да бъдеш забравен
- Приемливо ниво на сигурност
- Разходи за изчислителен процес, консумация на енергия,
- Стандарти за безопасност и прозрачност (лекарства, храни, козметика)
- Валидиране и изплащан на искове

Избор на платформа

- Определяне на целите и материалните потоци.
- Дефиниране на параметрите, които ще се измерват.
- Определяне на методите за измерване.
- Предлагане на метод за споделяне на данните според нормативната база.
- Анализ на ограничаващите условия за технологичния процес.

Подход

- Избор на платформа – разрешена, с оторизиран достъп, хибридна
 - Кой ще има достъп до системата
- Особености на данните – публична или частна
- Кой ще модифицира данните - публични, частни, хибридни или консорциумни
- Цена
- Скорост на транзакция
- Потребявана електроенергия
- Възможност за поддръжка
- Тенденции за преобразуване на платформата (например от частна в хибридна)

Blockchain приложения за енергийния сектор



- 18 производствени приложения
- 21 H2020 проекти, базирани на blockchains

ICARUS; CHARIOT; MERLON; SealedGRID; AGILE IOT; Vicinity; ChainRider; EFICHAIN; FlexHub; PARITY; MOPO; iDesignRES; PIONEER; OSMOSE; Smart4RES; Kairos@C; PLATONE, FARCROS; BD4OPEM; INTERRACE; SecureChain

- Pilot blockchains за енергийно моделиране



Blockchain приложения в сектор Застраховане

 NXM NXM

 inSure DeFi SURE

 Wrapped NXM WNXM

 Uno Re UNO

 Etherisc DIP Token DIP

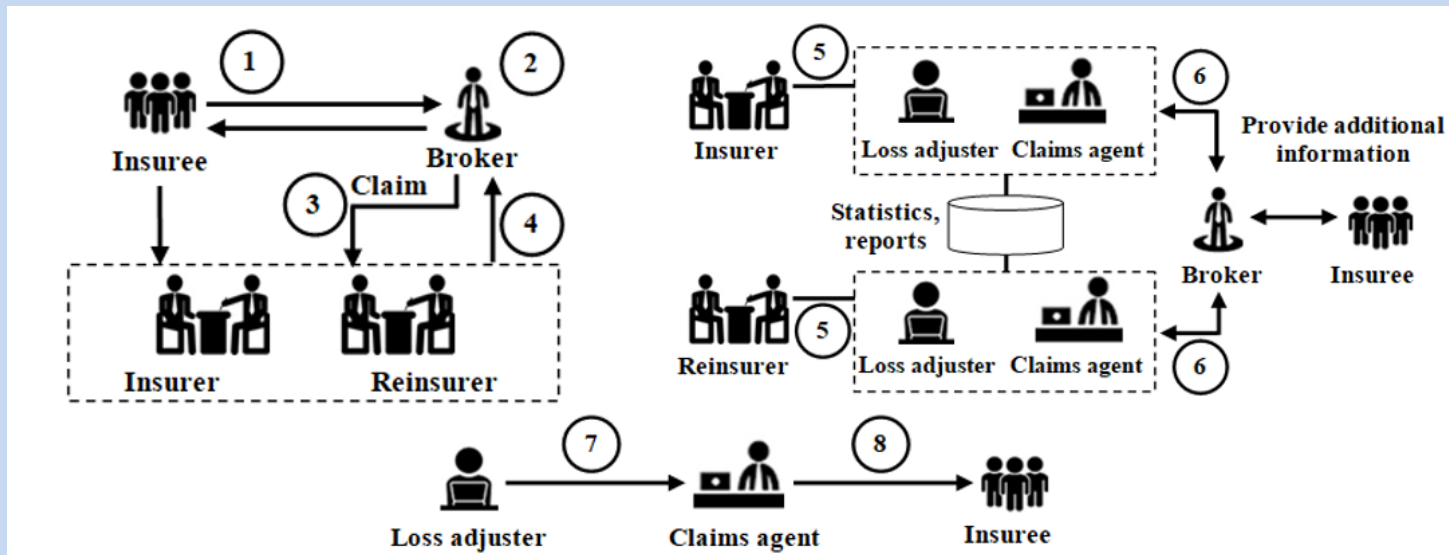
 YAM V3 YAM

 InsurAce INSUR

 Razor Network RAZOR

- 28 blockchains
- Подаването на искове е опростено и автоматизирано.
- Нуждата от брокери отпада.
- Намалява необходимото време за обработка на иск.
- Експертът не трябва да преглежда всеки иск (освен в специфични ситуации).
- Застрахователят има възможности за разкриване на потенциални опити за измама.
- Процесът на плащане за загуба е автоматизиран от интелигентния договор.
- Това елиминира необходимостта от посредник за искове.

Традиционно застраховане



Публікувано в:

1. Veneta Aleksieva, Hristo Valchanov and Anton Huliyan, Application of Smart Contracts based on Ethereum Blockchain for the Purpose of Insurance Services, 8-9.11.2019, Varna, BIA 2019, p. 69-72, ISBN 978-1-7281-4754-3, IEEE Catalog number: CFP19U85-ART, DOI: 10.1109/BIA48344.2019.8967468
2. V. Aleksieva, H. Valchanov and A. Huliyan, "Implementation of Smart-Contract, Based on Hyperledger Fabric Blockchain," 2020 21st International Symposium on Electrical Apparatus & Technologies (SIELA), Bourgas, Bulgaria, 2020, pp. 1-4, ISBN: 978-1-7281-4346-0, doi: 10.1109/SIELA49118.2020.9167043.
3. V. Aleksieva, H. Valchanov and A. Huliyan, "Smart Contracts based on Private and Public Blockchains for the Purpose of Insurance Services," 2020 International Conference Automatics and Informatics (ICAI), Varna, Bulgaria, 2020, pp. 1-4, doi: 10.1109/ICAI50593.2020.9311371, ISBN:978-1-7281-9309-0.

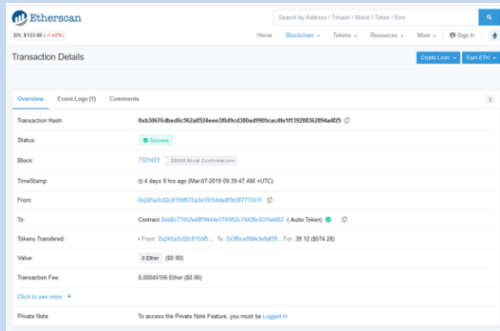
The diagram illustrates a smart contract-based insurance claim settlement process. It is divided into two main parts. The left part shows the initial setup and claim submission, while the right part shows the final settlement process.

Left Part: Initial Setup and Claim Submission

- Insuree:** Represented by a group of people icon. They initiate the process (Step 1) and receive information (Step 2).
- Smart asset:** Represented by a server icon. It stores **Insuree information, Claim history, Coverage period, Loss submission detail**. It interacts with the Insuree (Steps 2 and 3) and the Smart contract (Step 4).
- Smart contract:** Represented by a document icon with a gear. It receives data from the Smart asset (Step 4) and sends it to the Insurer/Reinsurer (Step 5).
- Insurer/Reinsurer:** Represented by a person at a desk icon. They receive the claim data (Step 5) and send it back to the Smart contract (Step 6).
- Statistics, reports:** A database icon that receives data from the Smart contract (Step 4) and sends it back (Step 5).

Right Part: Final Settlement

- Loss adjuster:** Represented by a person at a laptop icon. They receive the claim data (Step 7) and send it back to the Smart contract.
- Smart contract:** It receives the data from the Loss adjuster (Step 7) and sends it back to the Insuree for settlement.
- Insuree:** Represented by a group of people icon. They receive the settlement (Step 7).



metasploit.meterpreter

No.	Time	Source	Destination	Protocol	Length
254	12.461642	192.168.0.0-2	34.202.251.153	TLSv1-2	
255	12.461598	192.168.0.0-2	34.202.251.153	TLSv1-2	
256	12.461375	192.168.0.0-2	34.202.251.153	TLSv1-2	
257	12.461385	192.168.0.0-2	34.202.251.153	TLSv1-2	
258	12.460436	192.168.0.0-2	34.202.251.153	TLSv1-2	
259	12.460467	192.168.0.0-2	34.202.251.153	TLSv1-2	
260	12.460629	192.168.0.0-2	34.202.251.153	TLSv1-2	
261	12.460694	192.168.0.0-2	34.202.251.153	TLSv1-2	
262	12.460804	192.168.0.0-2	34.202.251.153	TLSv1-2	
263	12.460872	192.168.0.0-2	34.202.251.153	TLSv1-2	
264	12.460852	151.181.14.49	192.168.0.0-2	TLSv1-2	
266	12.781564	151.181.14.49	192.168.0.0-2	TLSv1-2	
268	12.714182	151.181.14.49	151.181.14.49	TLSv1-2	
270	12.751178	151.181.14.49	192.168.0.0-2	TLSv1-2	
283	12.630373	34.202.251.153	192.168.0.0-2	TLSv1-2	
285	12.637342	34.202.251.153	192.168.0.0-2	TLSv1-2	
287	12.444013	34.202.251.153	192.168.0.0-2	TLSv1-2	
289	12.444462	34.202.251.153	192.168.0.0-2	TLSv1-2	
290	12.445987	34.202.251.153	192.168.0.0-2	TLSv1-2	
292	12.445992	34.202.251.153	192.168.0.0-2	TLSv1-2	
293	12.446639	34.202.251.153	192.168.0.0-2	TLSv1-2	

369

Details for 34.202.251.153

IP: 34.202.251.153
 Decimal: 583728025
 Hostname: ec2-34-202-251-153.compute-1.amazonaws.com
 ASN: 14618
 ISP: Amazon.com
 Organization: Amazon.com
 Services: None detected
 Type: **Corporate**
 Assignment: **Static IP**
 Blacklist: [Click to Check Blacklist Status](#)
 Country: North America
 Country: United States
 State/Region: Virginia
 City: Ashburn
 Latitude: 39.0481 (39° 2' 53.16"N)
 Longitude: -77.4728 (77° 28' 22.08"W)
 Postal Code: 20149

```
lecbooks-MacBook-Air:AutoCoin nissmo$ truffle test
Using network 'development'.

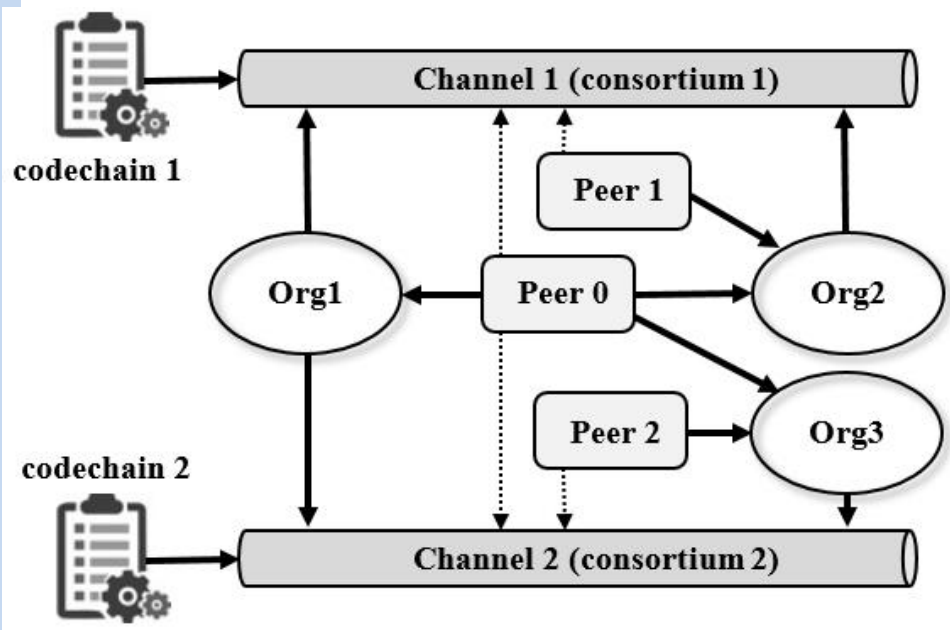
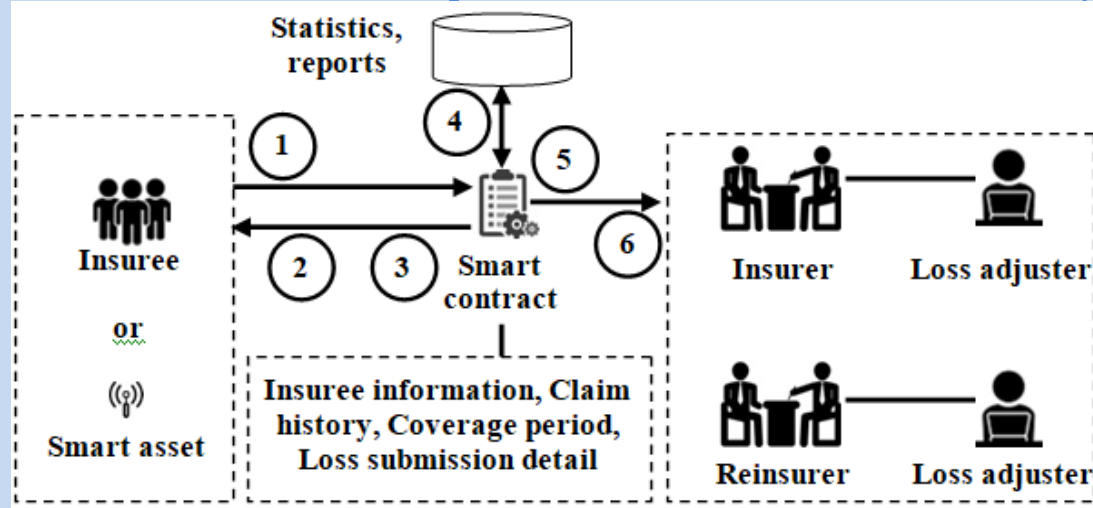
Contract: AutoCoin
  ✓ initializes the contract with the correct values (88ms)
  ✓ sets the total supply upon deployment (42ms)
  ✓ transfers token ownership (303ms)
  ✓ approves tokens for delegated transfer (155ms)
  ✓ handles delegated token transfers (692ms)

5 passing (1s)

lecbooks-MacBook-Air:AutoCoin nissmo$
```

13

Застраховане с HyperLedger



Канали:

- за консорциум 1 (Канал 1) на компания Org1 (застрахователна компания) и компания Org2 (брокер 1),
- за консорциум 2 (Канал 2) на компания Org1 и компания Org3 (брокер 2).

Всеки канал има свой собствен блокчейн, както и интелигентни договори, които работят самостоятелно с него.

Всеки консорциум има двама участници.

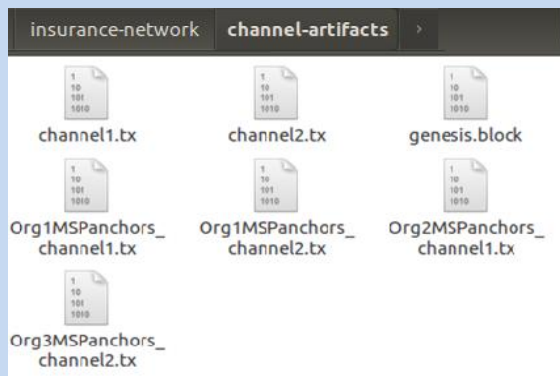
Двата канала работят паралелно един с друг и не са видими за участниците извън разрешените от правилата на консорциума.

- компания Org1 има свой собствен партньор Peer0, който участва в два консорциума, може да обслужва и двата канала. Той има отделни копия на двата блокчейна, без да бъдат компрометирани, благодарение на криптографските алгоритми на блокчейна и правилата за достъп, описани във всеки от консорциумите.
- В имплементираната бизнес логика е възможно клиентът да смени своя брокер. Това означава, че неговата полица трябва да се премести от един канал в друг канал.

Застраховане с HyperLedger

```
nissmo@nissmo-VirtualBox:~/fabric-samples/insurance-network$ ../bin/configtxgen -profile
SampleMultiNodeEtcdRaft -channelID insur-sys-channel -outputBlock ./channel-artifacts/g
enesis.block
2019-12-31 21:13:54.549 EET [common.tools.configtxgen] main -> INFO 001 Loading configur
ation
2019-12-31 21:13:54.583 EET [common.tools.configtxgen.localconfig] completeInitializatio
n -> INFO 002 orderer type: etcdraft
2019-12-31 21:13:54.583 EET [common.tools.configtxgen.localconfig] completeInitializatio
n -> INFO 003 Orderer.EtcdRaft.Options unset, setting to tickInterval:"500ms" election_
tick:10 heartbeat_tick:1 max_inflight_blocks:5 snapshot_interval_size:20971520
2019-12-31 21:13:54.583 EET [common.tools.configtxgen.localconfig] Load -> INFO 004 Load
ed configuration: /home/nissmo/fabric-samples/insurance-network/configtx.yaml
2019-12-31 21:13:54.622 EET [common.tools.configtxgen.localconfig] completeInitializatio
n -> INFO 005 orderer type: solo
2019-12-31 21:13:54.622 EET [common.tools.configtxgen.localconfig] LoadTopLevel -> INFO
006 Loaded configuration: /home/nissmo/fabric-samples/insurance-network/configtx.yaml
2019-12-31 21:13:54.622 EET [common.tools.configtxgen.encoder] NewOrdererGroup -> WARN 0
07 Default policy emission is deprecated, please include policy specifications for the o
rderer group in configtx.yaml
2019-12-31 21:13:54.623 EET [common.tools.configtxgen] doOutputBlock -> INFO 008 Generat
ing genesis block
2019-12-31 21:13:54.624 EET [common.tools.configtxgen] doOutputBlock -> INFO 009 Writing
genesis block
```

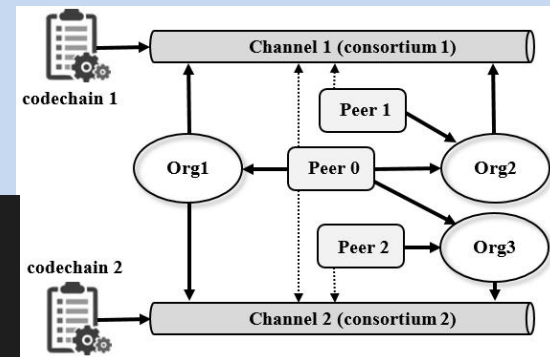
• Genesis block



```
services:
peer-base:
  image: hyperledger/fabric-peer:$IMAGE_TAG
  environment:
    - CORE_VM_ENDPOINT=unix:///host/var/run/docker.sock
    # the following settings starts chaincode containers on the same
    # bridge network as the peers
    - https://docs.docker.com/compose/networking/
    - CORE_VM_DOCKER_HOSTCONFIG_NETWORKMODE=${COMPOSE_PROJECT_NAME}.byfn
    - FABRIC_LOGGING_SPEC=INFO
    # FABRIC LOGGING SPEC=DEBUG
    - CORE_PEER_TLS_ENABLED=true
    - CORE_PEER_GOSSIP_USELEADERELECTION=true
    - CORE_PEER_GOSSIP_ORGLEADER=false
    - CORE_PEER_PROFILE_ENABLED=true
    - CORE_PEER_TLS_CERT_FILE=/etc/hyperledger/fabric/tls/server.crt
    - CORE_PEER_TLS_KEY_FILE=/etc/hyperledger/fabric/tls/server.key
    - CORE_PEER_TLS_ROOTCERT_FILE=/etc/hyperledger/fabric/tls/ca.crt
  working_dir: /opt/gopath/src/github.com/hyperledger/fabric/peer
  command: peer node start

orderer-base:
  image: hyperledger/fabric-orderer:$IMAGE_TAG
  environment:
    - FABRIC_LOGGING_SPEC=INFO
    - ORDERER_GENERAL_LISTENADDRESS=0.0.0.0
    - ORDERER_GENERAL_GENESISMETHOD=file
    - ORDERER_GENERAL_GENESISFILE=/var/hyperledger/orderer/orderer.genesis.block
    - ORDERER_GENERAL_LOCALMSPID=OrdererMSP
    - ORDERER_GENERAL_LOCALMSPDIR=/var/hyperledger/orderer/msp
    # enabled TLS
    - ORDERER_GENERAL_TLS_ENABLED=true
    - ORDERER_GENERAL_TLS_PRIVATEKEY=/var/hyperledger/orderer/tls/server.key
    - ORDERER_GENERAL_TLS_CERTIFICATE=/var/hyperledger/orderer/tls/server.crt
    - ORDERER_GENERAL_TLS_ROOTCAS=[/var/hyperledger/orderer/tls/ca.crt]
    - ORDERER_KAFKA_TOPIC_REPLICATIONFACTOR=1
    - ORDERER_KAFKA_VERBOSE=true
    - ORDERER_GENERAL_CLUSTER_CLIENTCERTIFICATE=/var/hyperledger/orderer/tls/server.crt
    - ORDERER_GENERAL_CLUSTER_CLIENTPRIVATEKEY=/var/hyperledger/orderer/tls/server.key
    - ORDERER_GENERAL_CLUSTER_ROOTCAS=[/var/hyperledger/orderer/tls/ca.crt]
  working_dir: /opt/gopath/src/github.com/hyperledger/fabric
  command: orderer
```

peer-base.yaml



```
type PolicyContract struct{
}

type Policy struct{
  Id string
  Type string
  Prem float32
  Owner Owner
  Property Property
  CreatedAt time.Time
}

type Owner struct{
  Name string
  FirstName string
  Gender string
  Address string
}

type Property struct{
  Type string
  Value string
}
```

• Artifacts for channels and genesis block.

Objects and attributes

Застраховане с HyperLedger

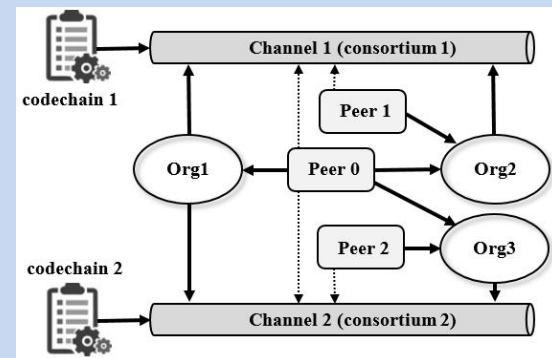
HYPERLEDGER EXPLORER DASHBOARD NETWORK BLOCKS TRANSACTIONS CHAINCODES CHANNELS channel1

Peer Name	Request Url	Peer Type	MSPID	Ledger Height		
				High	Low	Unsig
peer0.org1.example.com	peer0.org1.example.com:7051	PEER	Org1MSP	0	3	tru
peer0.org2.example.com	peer0.org2.example.com:9051	PEER	Org2MSP	0	3	tru
orderer.example.com	orderer.example.com:7050	ORDERER	OrdererM...	-	-	-

- Peers and orderer node

Block Details		Block Details	
Channel name:	channel1	Channel name:	channel1
Block Number	0	Block Number	1
Created at	2020-01-02T09:43:59.000Z	Created at	2020-01-02T12:43:27.000Z
Number of Transactions	1	Number of Transactions	1
Block Hash	d149cb3fa4ac12309e599d6718480d3acd4b7	Block Hash	e7b0a42970e890a4cb5e327579d71e662a83
Data Hash	82cf94b2203a2347c30a14135cec2c95793af	Data Hash	c0c132d2b293d76540acefc01790eb2a8d27f
Prehash		Prehash	d149cb3fa4ac12309e599d6718480d3acd4b7

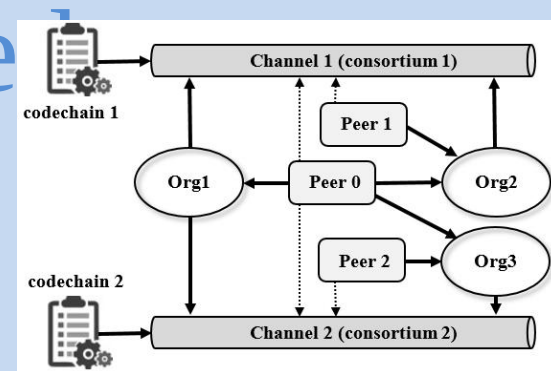
- created blocks



Застраховане с HyperLe

```
peer chaincode invoke -o orderer.example.com:7050 --tls true --cafile
/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/
orderer.example.com/msp/tlsacerts/tlsca.example.com-cert.pem -C $CHANNEL_NAME -n mycc -
root@ce651e966f64:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer chaincode invoke -o
orderer.example.com:7050 --tls true --cafile
/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/
orderer.example.com/msp/tlsacerts/tlsca.example.com-cert.pem -C $CHANNEL_NAME -n mycc --
peerAddresses peer0.org1.example.com:7051 --tlsRootCertFiles
/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/peers/p
eer0.org1.example.com/tls/ca.crt -c '{"Args":["CreatePerson","Anton","Huliyar","Male", "grad
Varna"]}'
2020-01-08 21:08:05.330 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 04c Chaincode
invoke successful. result: status:200
root@ce651e966f64:/opt/gopath/src/github.com/hyperledger/fabric/peer#
```

- Script and result in console for creation of a person in the channel



The screenshot shows the Hyperledger Explorer interface. The top navigation bar includes Dashboard, Network, Blocks, Transactions, Chaincodes, and Channels. The Transactions tab is selected, and a search filter is applied for the channel 'channel1'. The search results show a transaction with the ID 'a5eb9d360777c71706320b5294a44321416da7da9e73a04c6b31197bf0d39987'. The transaction details are displayed below, showing the Transaction ID, Validation Code (VALID), Payload Proposal Hash, Creator MSP (Org1MSP), Endoser (Org1MSP), Chaincode Name (mycc), Type (ENDORSE_TRANSACTION), and Time (2020-01-08T21:08:05.323Z). The transaction is a successful 'set' operation for the 'mycc' chaincode, creating a new person record.

Transaction ID	Validation Code	Payload Proposal Hash	Creator MSP	Endoser	Chaincode Name	Type	Time
a5eb9d360777c71706320b5294a44321416da7da9e73a04c6b31197bf0d39987	VALID	38bafb6bd041666fe9b91e2d61e839e44da0c2bf922205c6df8327de47bfbd46	Org1MSP	{Org1MSP}	mycc	ENDORSE_TRANSACTION	2020-01-08T21:08:05.323Z

Transaction Details

- Transaction ID: a5eb9d360777c71706320b5294a44321416da7da9e73a04c6b31197bf0d39987
- Validation Code: VALID
- Payload Proposal Hash: 38bafb6bd041666fe9b91e2d61e839e44da0c2bf922205c6df8327de47bfbd46
- Creator MSP: Org1MSP
- Endoser: {Org1MSP}
- Chaincode Name: mycc
- Type: ENDORSE_TRANSACTION
- Time: 2020-01-08T21:08:05.323Z
- Reads: root: [] 2 items
- Writes: root: [] 2 items
 - 0: {} 2 keys
 - 1: {} 2 keys
 - chaincode: "mycc"
 - set: [] 1 item
 - 0: {} 3 keys
 - key: "user-IBZgaiCMRAJWWhTHetcuAxxhKQFDaFp"
 - is_delete: false
 - value: [{"Name": "Anton", "FirstName": "Huliyar", "Gender": "Male", "Address": "grad Varna"}]

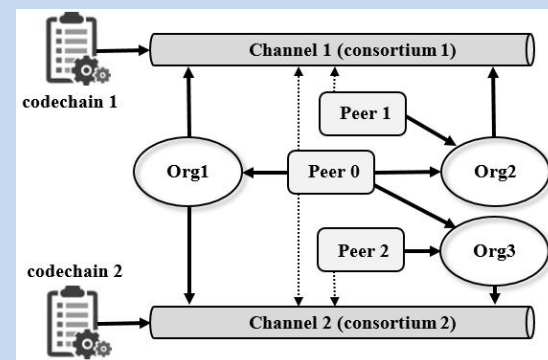
- Result in transaction for creation of a person in the channel

Застраховане с HyperLedger

```
peer chaincode invoke -o orderer.example.com:7050 --tls true --cafile
/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/
orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C $CHANNEL_NAME -n mycc --
peerAddresses peer0.org1.example.com:7051 --tlsRootCertFiles
/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/peers/p
eer0.org1.example.com/tls/ca.crt -c '{"Args":["queryOwnerByName", "Ivan"]}'
```

```
2020-01-08 22:16:31.372 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 04c
Chaincode invoke successful. result: status:200 payload:"[daaaaaa{"Key":\\"owner-lBzgba\\", \\"Record\\":{"Address\\":\\"grad Varna\\",\\"FirstName\\":\\"Ivanov\\",
\\"Gender\\":\\"Male\\",\\"Name\\":\\"Ivan\\"}}]"
```

- Script and result in console for the search of a person on the Blockchain



Transaction Details	
Transaction ID:	b3562a055e904373a1d9b4cd988c78cc846831d290f0824b4d9da6fc2a75fc2c
Validation Code:	VALID
Payload Proposal Hash:	429be19ba3d9cccd3f179ccd722b655986c6ea93556bcf29080f25ed0cd3bfd5
Creator MSP:	Org1MSP
Endoser:	Org1MSP
Chaincode Name:	mycc
Type:	ENDORSER_TRANSACTION
Time:	2020-01-08T22:16:31.358Z
Reads:	- root: [] 2 items 0: {} 2 keys 1: {} 2 keys - chaincode: "mycc" - set: [] 1 item 0: {} 2 keys - key: "owner-lBzgba" - version: {} 2 keys - block_num: "42" - tx_num: "0"

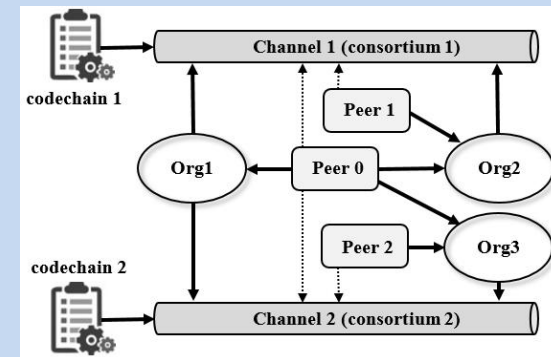
- Result in transaction for search of a person in the channel

Застраховане с HyperLedger

```
peer chaincode invoke -o orderer.example.com:7050 --tls true --cafile
/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/
orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C $CHANNEL_NAME -n mycc --
peerAddresses peer0.org1.example.com:7051 --tlsRootCertFiles
/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/peers/p
eer0.org1.example.com/tls/ca.crt -c '{"Args":["CreatePolicy", "Kirova", "Mirela", "female", "grad
Varna", "apartment", "41000", "fire"]}]'
```

```
2020-01-08 21:39:46.487 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 04c
Chaincode invoke successful. result: status:200 payload:"\n\"Successfully create
d person: owner-icMRAj\n\"Successfully created property: property-WwhTHc\n\"Succes
sfully created policy: policy-tcuAxxh"
```

- Script and result in console for creation of a policy

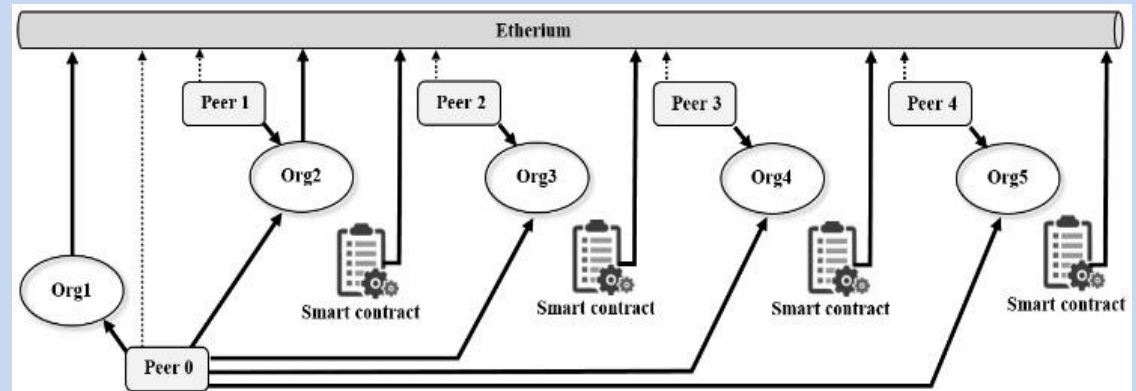
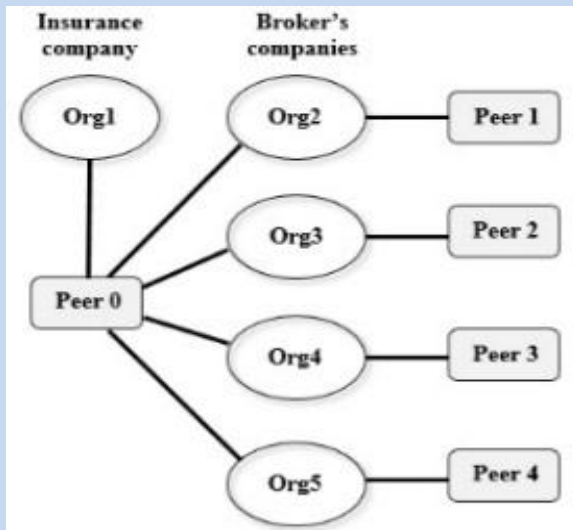


```
Endorser: { Org1mspr }
Chaincode Name: mycc
Type: ENDORSER_TRANSACTION
Time: 2020-01-08T21:39:46.482Z
Reads:
  root: [] 2 items
    0: {} 2 keys
    1: {} 2 keys
Writes:
  root: [] 2 items
    0: {} 2 keys
    1: {} 2 keys
      chaincode: "mycc"
      set: [] 3 items
        0: {} 3 keys
          key: "owner-icMRAj"
          is_delete: false
          value: "{\"Name\":\"Kirova\",\"FirstName\":\"Mirela\",\"Gender\":\"female\",\"Address\":\"grad Varna\"}"
        1: {} 3 keys
          key: "policy-tcuAxxh"
          is_delete: false
          value: "{\"Id\":\"\", \"Type\":\"fire\", \"Prem\":0, \"Owner\":{\"Name\":\"Kirova\",\"FirstName\":\"Mirela\",\"Gender\":\"female\", \"Address\":\"grad Varna\"}, \"Property\":{\"Type\":\"apartment\", \"Value\":\"41000\"}, \"CreatedAt\":\"2020-01-01T00:00:00Z\"}"
        2: {} 3 keys
          key: "property-WwhTHc"
          is_delete: false
          value: "{\"Type\":\"apartment\", \"Value\":\"41000\"}"
```

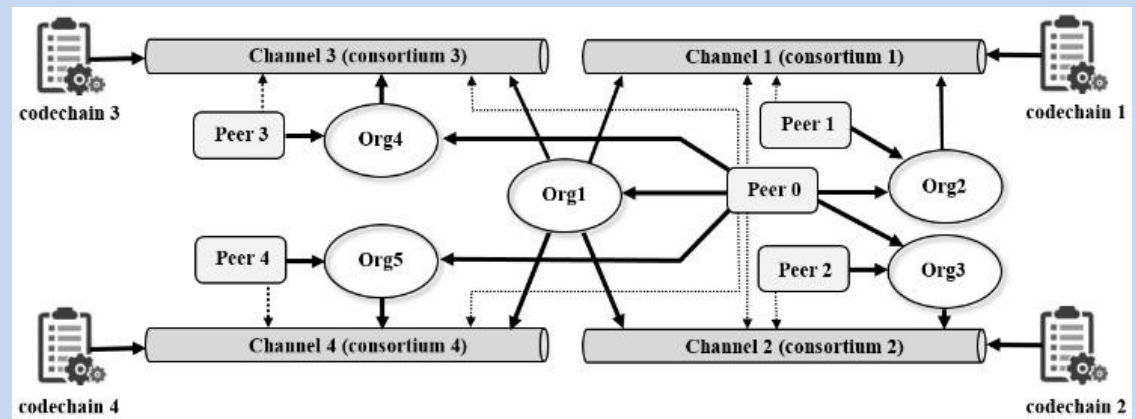
- Result in transaction for creation of a policy

Застраховане с HyperLedger и Ethereum

The implementation, based on Ethereum



The implementation, based on Hyperledger Fabric



Блокчейн приложения за Карго застраховки с IoT

 Travala.com AVA

 Ariva ARV

 TripCandy CANDY

 XcelToken Plus XLAB

 EvenCoin EVN

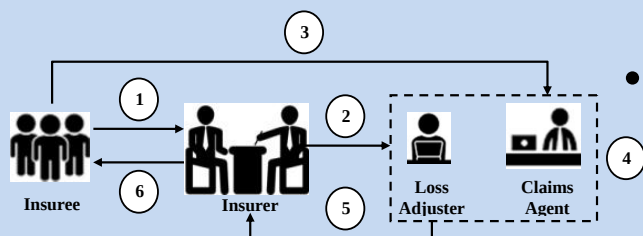
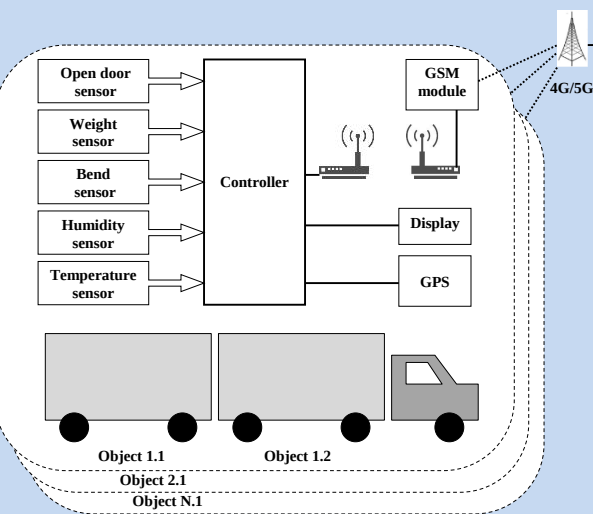
 Heron Asia HERON

 TravelNote TVNT

 Kemacoin KEMA

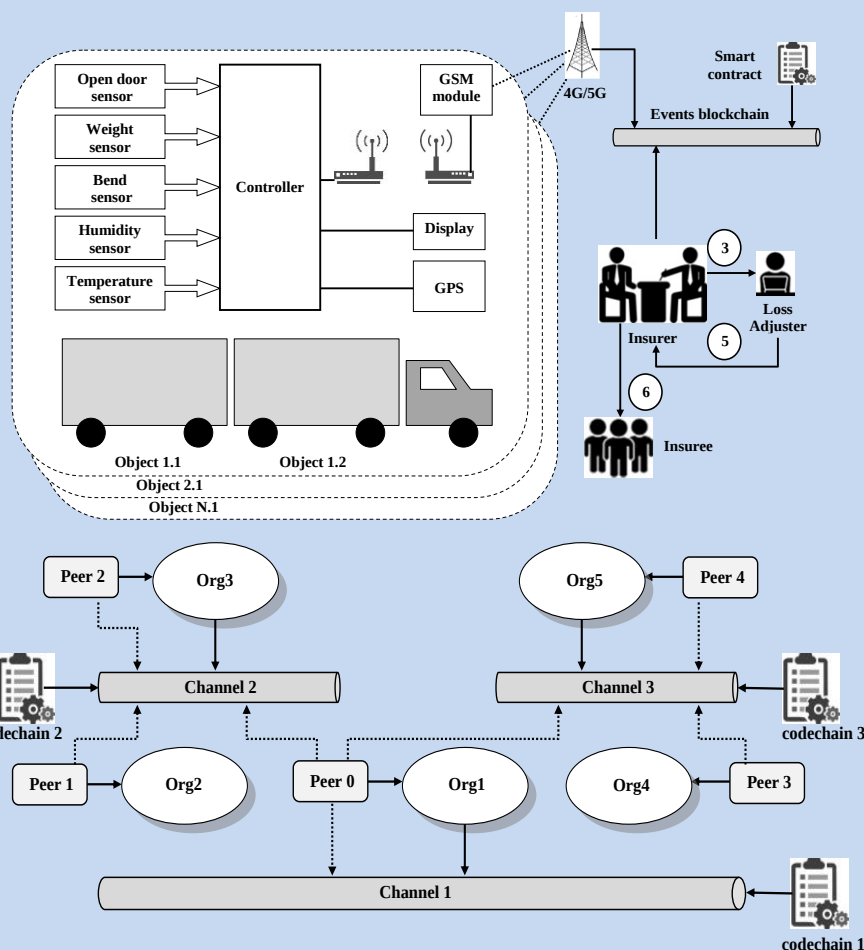
- 56 приложения
- информацията за щетата може да бъде изпратена директно от сензорите, монтирани в застрахования обект (контейнер, превозно средство) към автоматизирано приложение за обработка на иск.
- Интелигентният договор автоматично използва допълнителни източници (статистика, отчети), за да оцени иска и да изчисли щетите.
- Искът се обработва автоматично от интелигентен договор, базиран на зададена бизнес логика, като се използва информация, предоставена от застрахователя.
- Моделът предлага:
 - хронологично и прозрачно проследяване на данните от датчиците, разположени в контейнера/превозното средство,
 - изпраща навременни сигнали до заинтересованите страни, когато параметрите на датчиците превишават зададените прагове,
 - намалява документацията и оперативните разходи на застрахователя в случай на застрахователно събитие ,
 - елиминира възможността за измама,
 - подобрява удовлетвореността на клиентите при разглеждане на искове.

Традиционно Карго



- Информацията се изпраща по всяко време от сензорите към облачната/уеб базирана система, но само в една посока – от товара към системата за мониторинг.
- Данните от сензорите се изпращат през фиксирани периоди от време, които могат да се коригират в зависимост от вида на стоките.
- Съществува риск от подмяна на информацията, събрана и съхранявана от сензорите.
- Застрахованият трябва да попълни редица документи при подаване на застрахователна претенция.
- Отговорност на застрахования е да предостави доказателства за стойността на претърпените щети.
- Съществува риск от измама.
- Обработката на информацията е субективна, не е автоматизирана и експертите анализират много фактори при оценката на щетите. Възможно е различни експерти да определят различни дължими суми за една и съща щета.

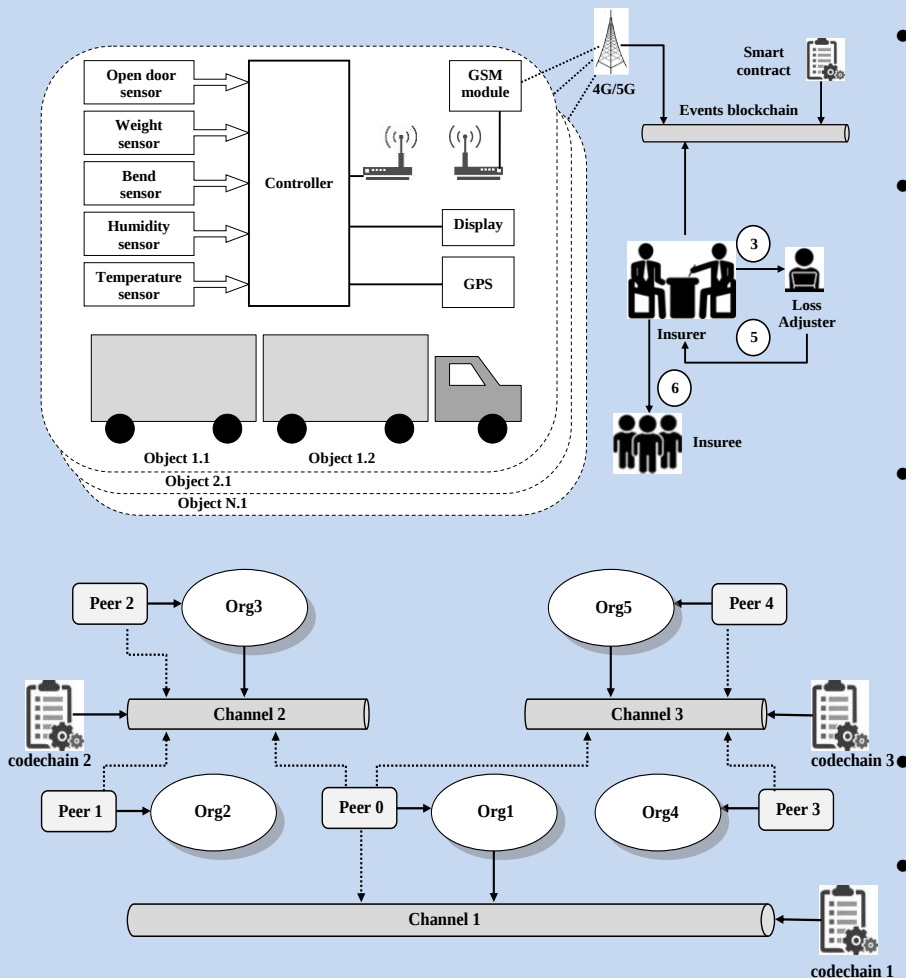
Карго с блокчейн и IoT



- Информация за щетите може да бъде изпратена директно от сензорите, инсталирани в застрахования обект (контейнер, превозно средство), до автоматизирано приложение за обработка на иск.
- Умният договор автоматично използва допълнителни източници (статистика, отчети), за да оцени иска и да изчисли щетите.

H.Valchanov, **V. Aleksieva**, Blockchain and IoT integration for smart transportation, EEPES'22, 22-24.-6.2022, Varna, Bulgaria, Journal of Physics: Conference Series, Volume 2339, International Conference on Electronics, Engineering Physics and Earth Science 2022 (EEPES 2022) 21/06/2022 - 24/06/2022 Varna, Bulgaria, Citation H Valchanov and V Aleksieva 2022 J. Phys.: Conf. Ser. 2339 012012, <https://doi.org/10.1088/1742-6596/2339/1/012012>

Карго с блокчейн и IoT



- Застрахователят е реер във всички канали (Peer 0), но той чете само данните от Канал 2 и Канал 3 и записва данни в Канал 1.
- В Канал 2 и Канал 3 се записват телеметричните данни от съответния обект (Peer 2 и Peer 4) с номера на обекта през определен интервал от време или при настъпване на събитие (стойностите надхвърлят праговете).
- Транспортните компании (Peer 1 и Peer 3), които са собственици на активите, имат достъп до Канал 2 и Канал 3, така че в случай на събитие да предприемат действия за минимизиране на щетите или предотвратяване на нежелано събитие. Застрахователните полици за съответните товари се записват в Канал 1.
- Интелигентният договор се грижи за проверката на застраховките и по инициатива на застрахователя с информация за настъпване на застрахователно събитие определя дължимата сума за застрахования.

Карго с блокчейн и IoT

CONTAINER ID	IMAGE	STATUS	PORTS	NAMES	COMMAND
1b0f79562884	dev-peer0.org1.example.com-mycc-1.0-384f11f484b9302df90b453200c7b25174305fca8f53f4e94d45ea3b6cab0ca9	Up 2 minutes		dev-peer0.org1.example.com-mycc-1.0	"chaincode -peer.add...
0556042elbe5	hyperledger/fabric-tools:latest	Up 2 minutes			"/bin/bash"
d17061014f9f	hyperledger/fabric-peer:latest	Up 2 minutes		peer0.org3.example.com	"peer node start"
24eade3d4fa4	hyperledger/fabric-peer:latest	Up 2 minutes	0.0.0.0:11051->11051/tcp	peer0.org1.example.com	"peer node start"
0bf549043fad	hyperledger/fabric-peer:latest	Up 2 minutes	0.0.0.0:7051->7051/tcp	peer0.org2.example.com	"peer node start"
81568abd2258	hyperledger/fabric-couchdb	Up 2 minutes	4369/tcp, 9100/tcp, 0.0.0.0:9984->5984/tcp	couchdb2	"tini -- /docker-ent...
148fcl11725c	hyperledger/fabric-orderer:latest	Up 2 minutes	0.0.0.0:9050->7050/tcp	orderer3.example.com	"orderer"
3a6dacacaffd	hyperledger/fabric-orderer:latest	Up 2 minutes	0.0.0.0:7050->7050/tcp	orderer.example.com	"orderer"
d446a85eef10	hyperledger/fabric-orderer:latest	Up 2 minutes	0.0.0.0:11050->7050/tcp	orderer5.example.com	"orderer"
acc307410954	hyperledger/fabric-couchdb	Up 2 minutes	4369/tcp, 9100/tcp, 0.0.0.0:5984->5984/tcp	couchdb0	"tini -- /docker-ent...
e395d663d107	hyperledger/fabric-couchdb	Up 2 minutes	4369/tcp, 9100/tcp, 0.0.0.0:7984->5984/tcp	couchdb1	"tini -- /docker-ent...
2666afb97675	hyperledger/fabric-orderer:latest	Up 2 minutes	0.0.0.0:10050->7050/tcp	orderer4.example.com	"orderer"
b1c8650a759d	hyperledger/fabric-orderer:latest	Up 2 minutes	0.0.0.0:8050->7050/tcp	orderer2.example.com	"orderer"

```
func (s *PolicyContract) Invoke(stub shim.ChaincodeStubInterface, args := stub.GetFunctionAndParameters())
if fc == "CreateCompany" {
    return s.CreateCompany(stub, args)
}
if fc == "CreatePolicy" {
    return s.CreatePolicy(stub, args)
}
if fc == "queryCompanyByName" {
    return s.queryCompanyByName(stub, args)
}
return shim.Error("Called function is not defined")
}
```

```
func (s *PolicyContract) CreateCompany(APIstub shim.ChaincodeStubInterface, args []string) ([]byte, error) {
    Id := "company-" + RandStringBytes(6)
    var company = Company{CargoCompany: args[0], VAT: args[1], Transport: args[2], Country: args[3]}
    UserBytes, _ := json.Marshal(company)
    APIstub.PutState(Id, UserBytes)
    return shim.Success(nil)
}
```

```
func (s *PolicyContract) CreatePolicy(APIstub shim.ChaincodeStubInterface, args []string) ([]byte, error) {
    var buffer bytes.Buffer

    Id_company := "company-" + RandStringBytes(6)
    Id_policy := "policy-" + RandStringBytes(6)

    var company = Company{CargoCompany: args[0], VAT: args[1], Transport: args[2], Country: args[3]}
    M_company, _ := json.Marshal(company)
    buffer.WriteString(M_company)
    buffer.WriteString(Id_policy)
    return shim.Success(buffer.Bytes())
}
```

```
type PolicyContract struct {
}

type Policy struct {
    Id string // Policy number
    Limit string // Limit of liability
    Premium string // Insurance bonus
    Company string // Cargo company
    ExpireDate string // Policy expiration date
    CreatedAt time.Time // Policy creation date
}

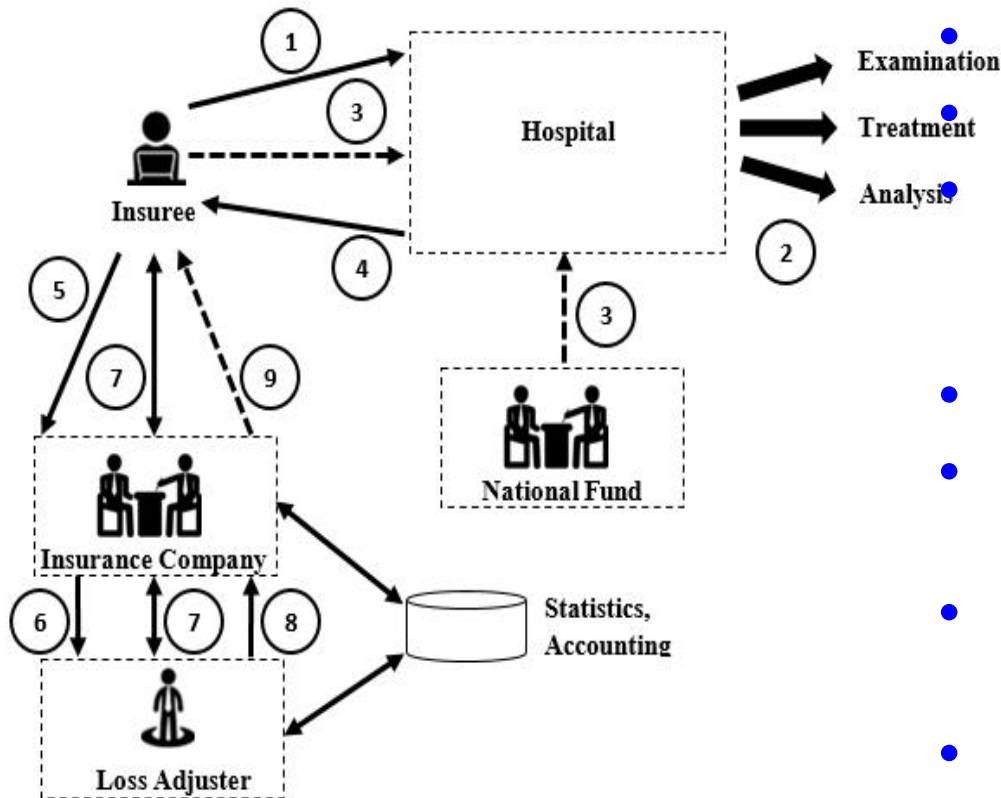
type Company struct {
    CargoCompany string // Name of the cargo company
    VAT string // Company VAT
    Transport string // Type of transport
    Country string // Company country
}
```

```
2022-04-14 07:10:24.067 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 04c Chaincode invoke successful. result: status:200 payload:{"OK>>>":{"Key":"company-baiCMR"},"Record":{"CargoCompany":"TransCargo","Country":"Bulgaria","Transport":"Truck"},"VAT":"12345"}}}
root@9b46df1006c2:/opt/gopath/src/github.com/hyperledger/fabric/peer#
```

ID	Channel Name	Blocks	Transactions	Timestamp
4	channel1	3	3	2022-04-12T11:44:25.000Z
3	channel2	3	3	2022-04-12T11:44:26.000Z

Channel name:	channel1
Block Number	0
Created at	2022-04-12T11:44:26.000Z
Number of Transactions	1
Block Hash	ec9a273580494367cc0bc8c26464403595eb14acbf282b375bee443f9cdf
Data Hash	a6424a90e66c016ddaa51bcb123f8ec288aa228e35ab18bd6d64337618b5d24
Prehash	44c5f2644e545d0eb734f0c0eb87db7618e531faa6ac3a724db7423e1e20

Блокчейн приложения за застраховка– Здраве- традиционно



- Examination Заболяване / злополука
- Treatment Лечение
- Analysis Когато лицето има задължителна здравна осигуровка – автоматично плащане
- Документи за лечение
- Иск за обезщетение към застрахователя
- Процес на проверка от застрахователя
- Искания/отговор за допълнителни доказателства
- заключение
- Плащане на застрахования

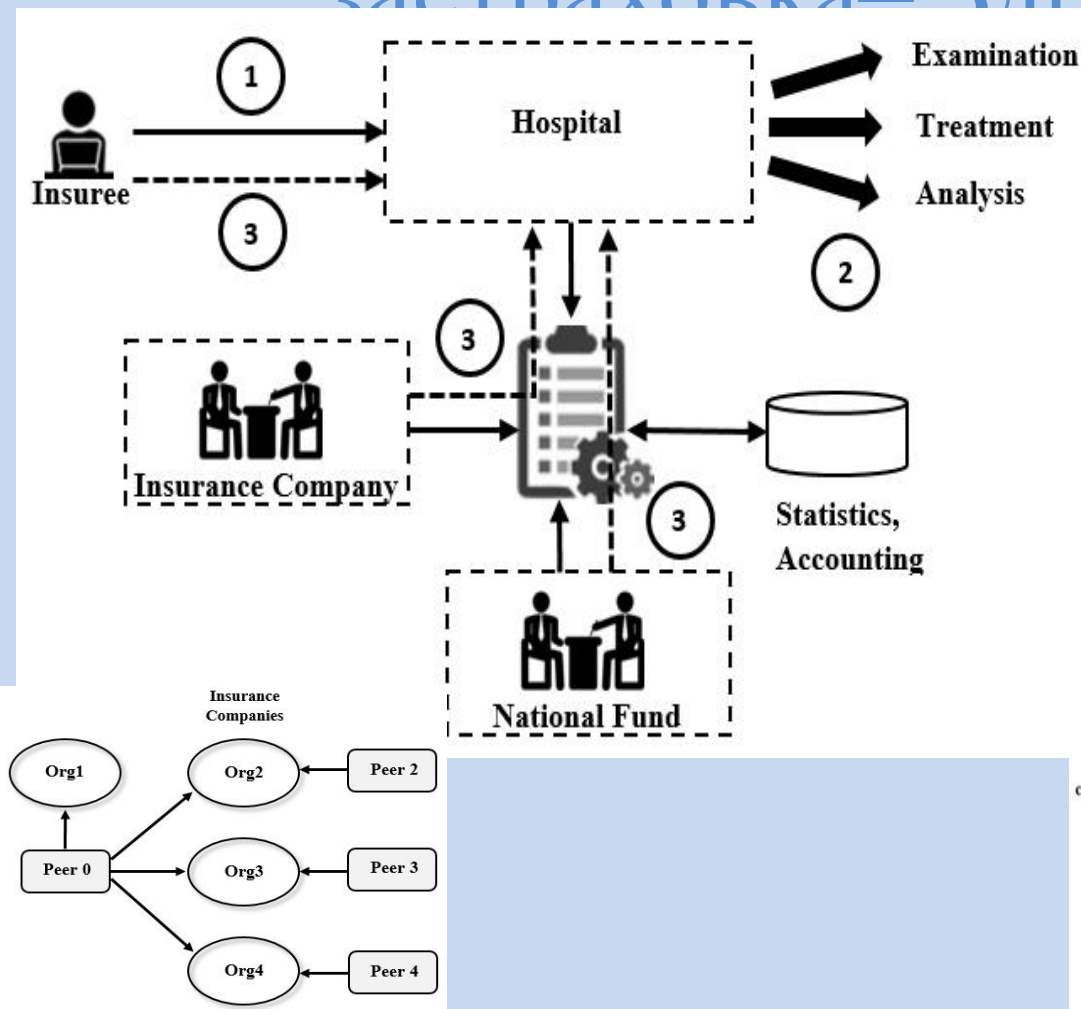
Публикувано в:

V. Aleksieva, H. Valchanov, Smart Contracts based on Hyperledger Fabric Blockchain for the Purpose of Health and Life Insurance Services, AIP, CIEES'21, 25-27.11.2021, Rouse, Bulgaria, ISBN: 978-0-7354-4375-4, vol. 2570, 18.08.2022, AIP Conference Proceedings 2570, 020002 (2022); <https://doi.org/10.1063/5.0099626>

9/1/2026

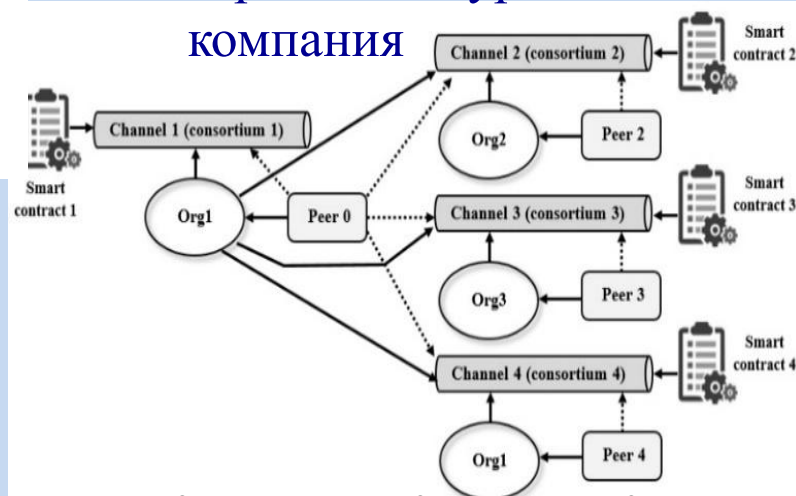
проф. д-р инж. Венета Алексиева

Блокчейн приложения за застраховка— Здраве с блокчейн



- Болест / злополука
- Лечение
- Когато лицето има задължителна здравна осигуровка – автоматично плащане от Националния фонд и здравноосигурителната

КОМПАНИЯ



Публикувано в:

V. Aleksieva, H. Valchanov, Smart Contracts based on Hyperledger Fabric Blockchain for the Purpose of Health and Life Insurance Services, AIP, CIEES'21, 25-27.11.2021, Rouse, Bulgaria, ISBN: 978-0-7354-4375-4, vol. 2570, 18.08.2022, AIP Conference Proceedings 2570, 020002 (2022); <https://doi.org/10.1063/5.0099626>

9/1/2026

проф. д-р инж. Венета Алексиева

27

Блокчейн приложения за застраховка— Здраве с блокчейн

channel-artifacts		
Name	Size	
channel1.tx	320	
channel2.tx	346	
channel3.tx	346	
channel4.tx	346	
genesis.block	37,5	
Org1MSPanchors_channel1.tx	282	
Org1MSPanchors_channel2.tx	282	
Org1MSPanchors_channel3.tx	282	
Org1MSPanchors_channel4.tx	282	
Org2MSPanchors_channel2.tx	282	
Org3MSPanchors_channel3.tx	282	
Org4MSPanchors_channel4.tx	282	

bc@bchain:~ ! update is a security update. Your Hardware Enablement Stack (HWE) is supported until April 2023. Last login: Sun May 16 10:37:25 2021 from 82.146.1.47 bc@bchain:~\$ docker ps -a						
CONTAINER ID	IMAGE	NAME	COMMAND	CREATED	STATUS	PORTS
f29a64b7556c	hyperledger/explorer:latest	explorer.mynetwork.com	"docker-entrypoint.s..."	21 hours ago	Up 21 hours	0.0.0.0
33b0b49131be	hyperledger/explorer-dh:latest	explorer-dh.mynetwork.com	"docker-entrypoint.s..."	21 hours ago	Up 21 hours (healthy)	5432/tcp
9306f6d602a0	dev-peer0.org2.example.com-myc-1.0-15b571b3ce45066b7ec74497da3b27e4e0df1345daf1951b94245ce09c42b	dev-peer0.org2.example.com-myc-1.0	"chaincode -peer.add..."	21 hours ago	Up 21 hours	
f94240ba25c7	dev-peer0.org1.example.com-myc-1.0-384f11f484b9302df90b43320c6fb25174305f0c8f53f4c94d45cc3b6cab0cc9	dev-peer0.org1.example.com-myc-1.0	"chaincode -peer.add..."	21 hours ago	Up 21 hours	
22e062ba6903	hyperledger/fabric-tools:latest	cli	"/bin/bash"	21 hours ago	Up 21 hours	
671940ba9516	hyperledger/fabric-peer:latest	peer0.org4.example.com	"peer node start"	21 hours ago	Up 21 hours	0.0.0.0
112051->112051/tcp	hyperledger/fabric-peer:latest	peer0.org4.example.com	"peer node start"	21 hours ago	Up 21 hours	0.0.0.0
3694dc54771c	hyperledger/fabric-peer:latest	peer0.org2.example.com	"peer node start"	21 hours ago	Up 21 hours	0.0.0.0
3051->3051/tcp	hyperledger/fabric-peer:latest	peer0.org2.example.com	"peer node start"	21 hours ago	Up 21 hours	0.0.0.0
36305cc3b00	hyperledger/fabric-peer:latest	peer0.org3.example.com	"peer node start"	21 hours ago	Up 21 hours	0.0.0.0
110051->110051/tcp	hyperledger/fabric-peer:latest	peer0.org3.example.com	"peer node start"	21 hours ago	Up 21 hours	0.0.0.0
b302406d4da	hyperledger/fabric-peer:latest	peer0.org1.example.com	"peer node start"	21 hours ago	Up 21 hours	0.0.0.0
7051->7051/tcp	hyperledger/fabric-couchdb	couchdb1	"tini -- /docker-ent..."	21 hours ago	Up 21 hours	4369/tcp
711fed3b3e7f	hyperledger/fabric-couchdb	couchdb1	"tini -- /docker-ent..."	21 hours ago	Up 21 hours	4369/tcp
p_9103/tcp, 0.0.0.0:6904->5954/tcp	hyperledger/fabric-orderer:latest	orderer1.example.com	"orderer"	21 hours ago	Up 21 hours	0.0.0.0
b764d92a33	hyperledger/fabric-orderer:latest	orderer1.example.com	"orderer"	21 hours ago	Up 21 hours	0.0.0.0
110950->7059/tcp	hyperledger/fabric-orderer:latest	orderer1.example.com	"orderer"	21 hours ago	Up 21 hours	0.0.0.0
8c978abebf0c	hyperledger/fabric-orderer:latest	orderer1.example.com	"orderer"	21 hours ago	Up 21 hours	0.0.0.0
3655->7050/tcp	hyperledger/fabric-orderer:latest	orderer1.example.com	"orderer"	21 hours ago	Up 21 hours	0.0.0.0
0a1f4e41d0	hyperledger/fabric-orderer:latest	orderer1.example.com	"orderer"	21 hours ago	Up 21 hours	0.0.0.0
10090->7059/tcp	hyperledger/fabric-orderer:latest	orderer1.example.com	"orderer"	21 hours ago	Up 21 hours	0.0.0.0
1028124fe116	hyperledger/fabric-couchdb	couchdb3	"tini -- /docker-ent..."	21 hours ago	Up 21 hours	4369/tcp
p_9100/tcp, 0.0.0.0:9984->5954/tcp	hyperledger/fabric-couchdb	couchdb3	"tini -- /docker-ent..."	21 hours ago	Up 21 hours	4369/tcp
37351b6c25d0	hyperledger/fabric-couchdb	couchdb3	"tini -- /docker-ent..."	21 hours ago	Up 21 hours	4369/tcp
3655->7050/tcp	hyperledger/fabric-couchdb	couchdb3	"tini -- /docker-ent..."	21 hours ago	Up 21 hours	4369/tcp
3331276bde0	hyperledger/fabric-peer:latest	peer1.org2.example.com	"peer node start"	21 hours ago	Up 21 hours	0.0.0.0
39051->9051/tcp	hyperledger/fabric-peer:latest	peer1.org2.example.com	"peer node start"	21 hours ago	Up 21 hours	0.0.0.0
5c4f5894797	hyperledger/fabric-peer:latest	peer1.org4.example.com	"peer node start"	21 hours ago	Up 21 hours	0.0.0.0
13081->13051/tcp	hyperledger/fabric-peer:latest	peer1.org4.example.com	"peer node start"	21 hours ago	Up 21 hours	0.0.0.0
2af3b044237	hyperledger/fabric-peer:latest	peer1.org3.example.com	"peer node start"	21 hours ago	Up 21 hours	0.0.0.0
11051->11051/tcp	hyperledger/fabric-peer:latest	peer1.org3.example.com	"peer node start"	21 hours ago	Up 21 hours	0.0.0.0
4d8131c0987c	hyperledger/fabric-orderer:latest	orderer1.example.com	"orderer"	21 hours ago	Up 21 hours	0.0.0.0
77050->7050/tcp	hyperledger/fabric-couchdb	couchdb2	"tini -- /docker-ent..."	21 hours ago	Up 21 hours	4369/tcp
83194256157b	hyperledger/fabric-couchdb	couchdb2	"tini -- /docker-ent..."	21 hours ago	Up 21 hours	4369/tcp
p_9103/tcp, 0.0.0.0:7934->5954/tcp	hyperledger/fabric-couchdb	couchdb2	"tini -- /docker-ent..."	21 hours ago	Up 21 hours	4369/tcp
ea4d00c8ffa6	hyperledger/fabric-couchdb	couchdb2	"tini -- /docker-ent..."	21 hours ago	Up 21 hours	4369/tcp
p_9103/tcp, 0.0.0.0:5994->5954/tcp	hyperledger/fabric-couchdb	couchdb2	"tini -- /docker-ent..."	21 hours ago	Up 21 hours	4369/tcp
bc@bchain:~\$						

```
type Policy struct{
    Id string           // Policy number
    Limit string        // Limit of liability
    Premium string      // Insurance bonus
    Owner Owner         // Owner
    ExpireDate string   // Policy expire date
    CreatedAt time.Time // Creation date
}

type Owner struct{
    Name string
    SurName string
    EGN string
}
```

```
func (s *PolicyContract) Invoke(stub shim.ChaincodeStubInterface) sc.Response {
    fc, args := stub.GetFunctionAndParameters()
    if fc == "CreatePerson" {
        return s.CreatePerson(stub, args)
    }
    if fc == "CreatePolicy" {
        return s.CreatePolicy(stub, args)
    }
    if fc == "queryOwnerByName" {
        return s.queryOwnerByName(stub, args)
    }
    return shim.Error("called function is not defined in the chaincode ")
}

func (s *PolicyContract) CreatePerson(APIstub shim.ChaincodeStubInterface, args []string) sc.Response {
    Id := "owner-" + RandStringBytes(6)
    var owner = Owner{Name: args[0], SurName: args[1], EGN: args[2]}
    UserBytes, _ := json.Marshal(owner)
    APIstub.PutState(Id, UserBytes)
    return shim.Success(nil)
}

func (s *PolicyContract) CreatePolicy(APIstub shim.ChaincodeStubInterface, args []string) sc.Response {
    var buffer bytes.Buffer

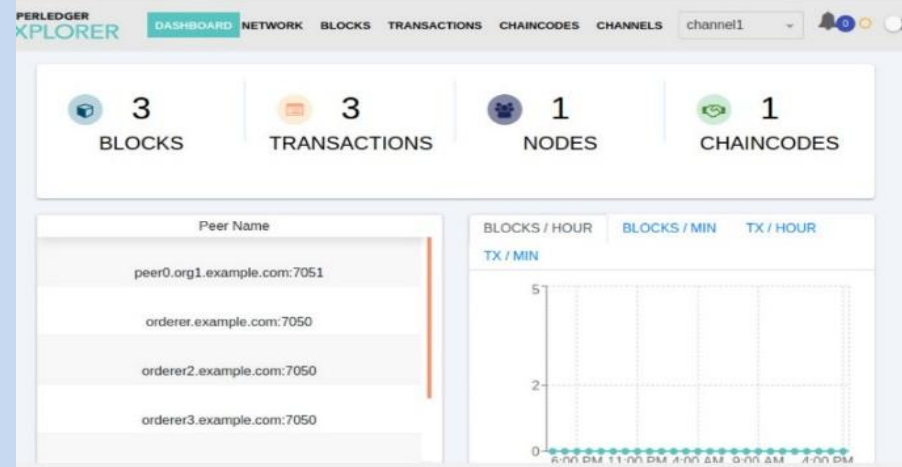
    Id_owner := "owner-" + RandStringBytes(6)
    Id_policy := "policy-" + RandStringBytes(6)

    var owner = Owner{Name: args[0], SurName: args[1], EGN: args[2]}
    M_owner, _ := json.Marshal(owner)
    err := APIstub.PutState(Id_owner, M_owner)
    if err != nil {
```

HYPERLEDGER EXPLORER DASHBOARD NETWORK BLOCKS TRANSACTIONS CHAINCODES CHA

Peer Name	Request Url	Peer Type	MSPID	High
peer0.org1.exa...	peer0.org1.exa...	PEER	Org1MSP	-
orderer.exampl...	orderer.exampl...	ORDERER	OrdererMSP	-
orderer2.exam...	orderer2.exam...	ORDERER	OrdererMSP	-
orderer3.exam...	orderer3.exam...	ORDERER	OrdererMSP	-
orderer4.exam...	orderer4.exam...	ORDERER	OrdererMSP	-

Previous Page 1 of 2 5 rows



Block Details

Channel name:	channel1
Block Number	0
Created at	2021-05-13T13:06:32.000Z
Number of Transactions	1
Block Hash	13bca1bd3c6092d637658d125114c8bfb64c3e184f6b0fa4f5d886a60c47346e
Data Hash	c0c5e7a02d020e7df148ae3e1b7c63970e978613bcd6d60b70453208b20e5f50
Prehash	

Block Details

Channel name:	channel1
Block Number	1
Created at	2021-05-13T13:06:34.000Z
Number of Transactions	1
Block Hash	27d3326f6419a8f1f8f034193c6d482f79502b490526a321519ddc9ae43793c
Data Hash	44579242951f6bd8d21b6d8b72eb8e042da1ca638cc6cf161ae7653e4c0cc6c
Prehash	13bca1bd3c6092d637658d125114c8bfb64c3e184f6b0fa4f5d886a60c47346e

Transaction Details

Transaction ID:	bfb546fbd007287c2b899005f2f511867e8ef6454e5f8dc9ac6c50128075222d
Validation Code:	VALID
Payload Proposal Hash:	47be47a06cd0a49e2780b2cc34131c4098f4aa2db86dc4850b7d7beddc45c362
Creator MSP:	Org1MSP
Endorser:	{ "Org1MSP" }
Chaincode Name:	Isscc
Type:	ENDORSER_TRANSACTION
Time:	2021-05-13T13:06:43.601Z
Reads:	- root: {} 1 item 0: {} 2 keys
Writes:	- root: {} 1 item 0: {} 2 keys

Transaction Details

Transaction ID:	cd6f0f6e823aeac4514c645b4c05d3afd97d6bd4cf9d0b0cb83460fb931f4ae
Validation Code:	
Payload Proposal Hash:	
Creator MSP:	OrdererMSP
Endorser:	{}
Chaincode Name:	
Type:	CONFIG
Time:	2021-05-13T13:06:34.000Z
Reads:	- root: {} 5 keys - version: 0 - groups: {} 1 key - values: {} 0 keys - policies: {} 0 keys - mod_policy: ""
Writes:	- root: {} 5 keys - version: 0 - groups: {} 1 key - values: {} 0 keys

Блокчейн приложения за IoT сектора

– Хоспитализация

 Travala.com AVA

 TripCandy CANDY

 Heron Asia HERON

 LockTrip LOC

 Emphy EPY

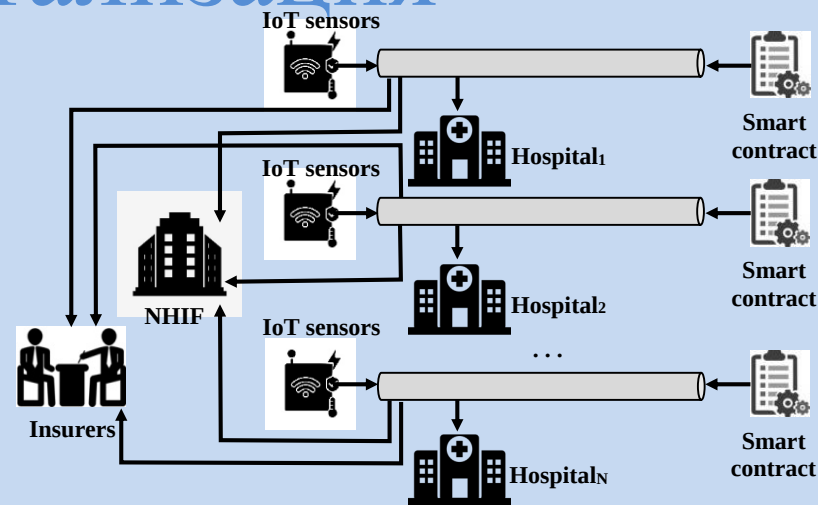
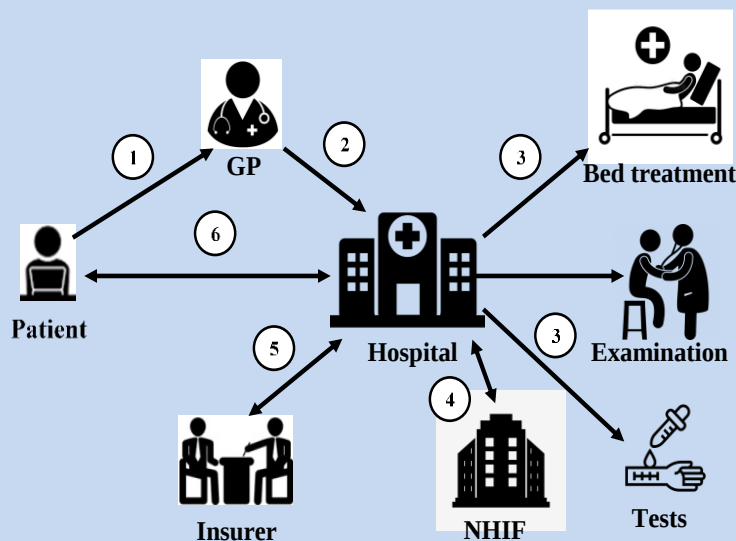
 Tripio TRIO

 Sharity \$SHARI

- 7 приложения
- традиционния модел, който има много недостатъци като:
 - Невъзможност за проследяване на дублиране на хоспитализирани пациенти – едновременно един пациент е хоспитализиран в две болници.
 - Невъзможност за проследяване на избора на клинични пътеки с по-добро финансиране, но неотговарящи на заболяването на пациента.
 - Невъзможност за спазване на максимален срок по клинична пътека.
 - Фиктивни хоспитализации.
 - Контрол се извършва само по предоставените документи от болницата или по сигнал.
 - НЗОК не следи дали всички хоспитализирани са реални.

Блокчейн приложения за IoT сектора

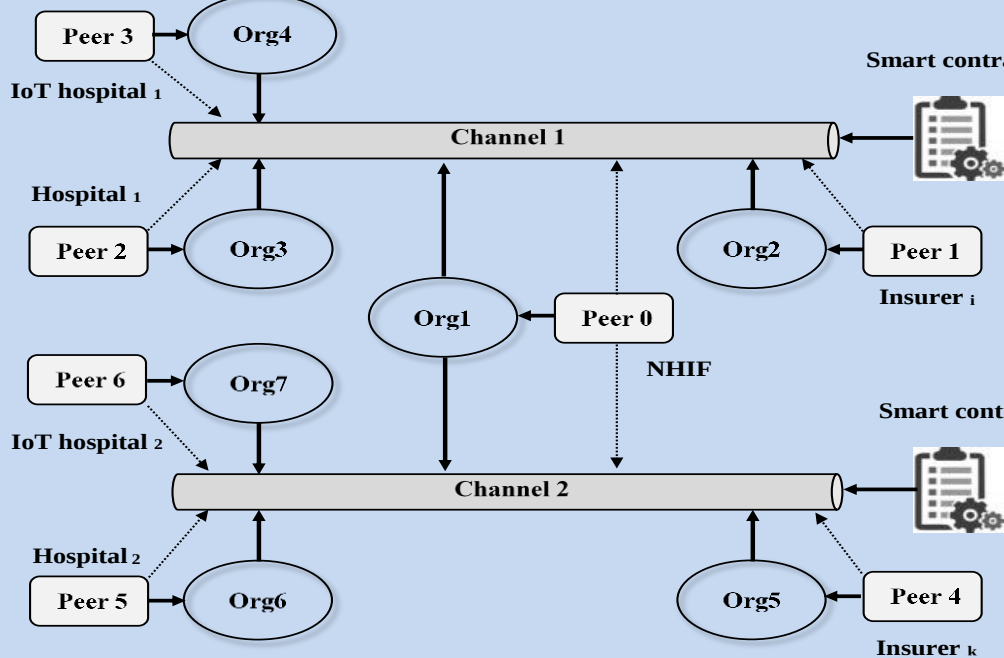
– Хоспитализация



- Моделът осигурява следене на заетостта на леглата в болниците чрез събиране на информация от сензори в леглата и от фитнес тракер за всеки пациент според номера на леглото.
- Събраната информация е полезна за болницата и за проследяване на здравословното състояние на пациента по време на лечението.
- Моделът предоставя на заинтересованите страни възможността да проследяват движенията на пациентите и да определя местоположението им, за да предотвратят фалшиви хоспитализации и неоправдани разходи.

Публикувано в:

H. Valchanov and V. Aleksieva, "Novel Model for Hospitalization Tracking based on Smart Contracts and IoT," 2022 International Conference Automatics and Informatics (ICAI), 2022, pp. 14–17, doi:10.1109/ICAI55857.2022.9959996



```
type SensorData struct{
    UsedBed string      // If the bed is used
    Pulse string        // Patient pulse
    Oxydation string    // Patient oxydation
    Location string     // Patient location
    CreatedAt time.Time // Sensor data creation time
    PID string         // Patient personal ID
}
```

```
func (s *HealthContract) CreateSensorData(APIStub shim.ChaincodeStubInterface,
    args []string) sc.Response {
    Id := "sensors-" + RandStringBytes(6)
    var sensors = SensorData{UsedBed: args[0], Pulse: args[1], Oxydation:
        args[2], Location: args[3], PID: args[4] }
    UserBytes, _ := json.Marshal(sensors)
    APIStub.PutState(Id, UserBytes)
    return shim.Success(nil)
}
```

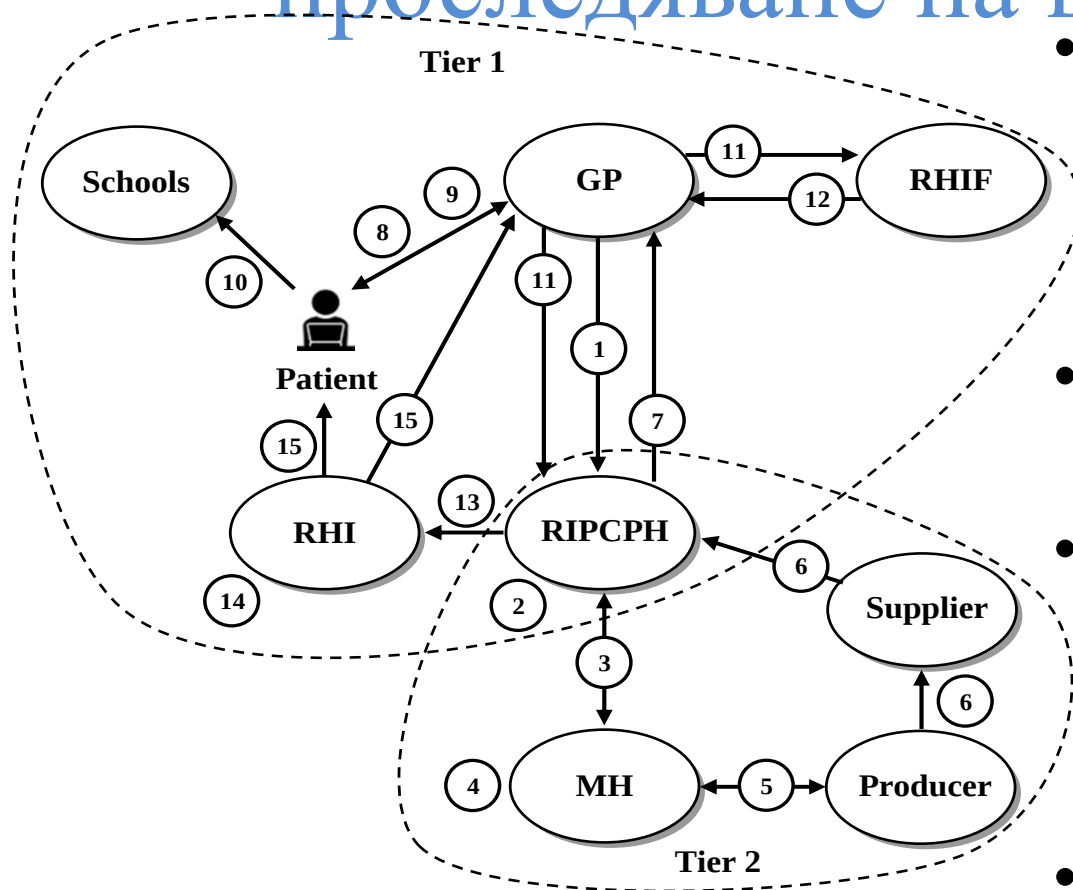
```
2022-06-29 13:17:49.790 UTC [msp.identity] Sign -> DEBU 04a Sign
    digest: 04588E14BC7B699C9E121B9C0818FD84F8F09A610A461AD443A7A
    51D9501036
2022-06-29 13:17:49.794 UTC [chaincodeCmd] chaincodeInvokeOrQue
    y -> DEBU 04b ESCC invoke result: version:1 response:<status:200
    > payload:"\n 212r\377\177\263\365\r\3467?\321\021\307\325a\34
    04ICmm\336R\257\0325*\274\250\367\304\207\022\306\001\n\260\001\
    022\025\n\0041sc\022\r\n\013\n\005smart\022\002\010\002\022\226
    \001\n\005smart\022\214\001\032\211\001\n\016sensors-HctcuA\032w
    {\\"UsedBed\\":\\"Yes\\",\\"Pulse\\":\\"63\\",\\"Oxydation\\":\\"94\\",\\"Loc
    ation\\":\\"Varna\\",\\"CreatedAt\\":\\"0001-01-01T00:00:00Z\\",\\"PID\\
    \":\\"111222333\\",\\"032\003\010\310\001\\",\\"014\022\005smart\032\0031.
    0\" endorsement:<endorser:\\n\007Org1MSP\022\252\006----BEGIN CE
    RTIFICATE-----\nMIICKDCCAC6gAwIBAgIQPoSvfm2Pin+ikKvJ38V9mJAKBggq
    wdKpUsVAdmaeNx15k05NWBLx5ydLAIIBBP3db\032\nvT6hpMsztXtS0aieSInlvcvFlm
    4QNFxLwrI2Vg==\n-----END CERTIFICATE-----\n\" signature:\"0D\002 N
    T>^\230H\242K\231\333kjAy0\035\311{\374\333\343\241.\314\205\243
    \361\374\343\244\237\002 <\216\t\233\000\367t\t\363\247\021T\
    212\357[\343qh n$ \365K\370\325)\214\341\276\t\" >
2022-06-29 13:17:49.795 UTC [chaincodeCmd] chaincodeInvokeOrQue
    y -> INFO 04c Chaincode invoke successful. result: status:200
```

```
\002\022\037\n\005smart\022\026\n\024\n\016sensors-HctcuA\022\00
2\010\017\032\250\001\010\310\001\032\242\001\0K>>>{\\"Key\\":\\"s
ensors-HctcuA\\", \\"Record\\":{\\"CreatedAt\\":\\"0001-01-01T00:00:00
Z\\",\\"Location\\":\\"Varna\\",\\"Oxydation\\":\\"94\\",\\"PID\\":\\"11122
333\\",\\"Pulse\\":\\"63\\",\\"UsedBed\\":\\"Yes\\",\\"}}\014\022\005smart
\032\0031.0\" endorsement:<endorser:\\n\007Org1MSP\022\252\006----
--BEGIN CERTIFICATE-----\nMIICKDCCAC6gAwIBAgIQPoSvfm2Pin+ikKvJ38
91196pD215wdKpUsVAdmaeNx15k05NWBLx5ydLAIIBBP3db\032\nvT6hpMsztXtS0aie
SInlvcvFlm4QNFxLwrI2Vg==\n-----END CERTIFICATE-----\n\" signature:\"0D\002
b\307\267\266v\362h4\2\241\306\037\2330if\361\325125\n
\233\365\221\2523\314\331\330\200\002 G\367\233\023\221\014\000\
201\230\310>qR\330\306\313\325\342\263U\035\274\035\200\nb\351
\304:\343\270\" >
2022-06-29 13:21:57.159 UTC [chaincodeCmd] chaincodeInvokeOrQue
y -> INFO 04c Chaincode invoke successful. result: status:200 pe
payload: \"[OK]>>>{\\"Key\\":\\"sensors-HctcuA\\", \\"Record\\":{\\"Created
At\\":\\"0001-01-01T00:00:00Z\\",\\"Location\\":\\"Varna\\",\\"Oxydation
\\":\\"94\\",\\"PID\\":\\"111222333\\",\\"Pulse\\":\\"63\\",\\"UsedBed\\":\\"Y
es\\",\\"}}\"
```

CONTAINER ID	PORTS
f17a787db515	
c9ab3e9da9d4	0.0.0.0:10051->10051/tcp
0b583ad50793	0.0.0.0:8051->8051/tcp
d58e4e4d1768	0.0.0.0:12051->12051/tcp
7fbaf0a2550	0.0.0.0:7051->7051/tcp
5fb66ccd7bab	0.0.0.0:8050->7050/tcp
5116f4bf458d	4369/tcp, 9100/tcp, 0.0.0.0:7984->5984/tcp
d967d02720fd	0.0.0.0:7050->7050/tcp
178ec2e0cd7a	4369/tcp, 9100/tcp, 0.0.0.0:6984->5984/tcp
0ab93fb9ba85	4369/tcp, 9100/tcp, 0.0.0.0:9984->5984/tcp
49d3f0c75916	0.0.0.0:9050->7050/tcp
e38e8b73bab7	0.0.0.0:10050->7050/tcp
0144bbadd52c	0.0.0.0:11050->7050/tcp
6176b7efb7f3	4369/tcp, 9100/tcp, 0.0.0.0:5984->5984/tcp

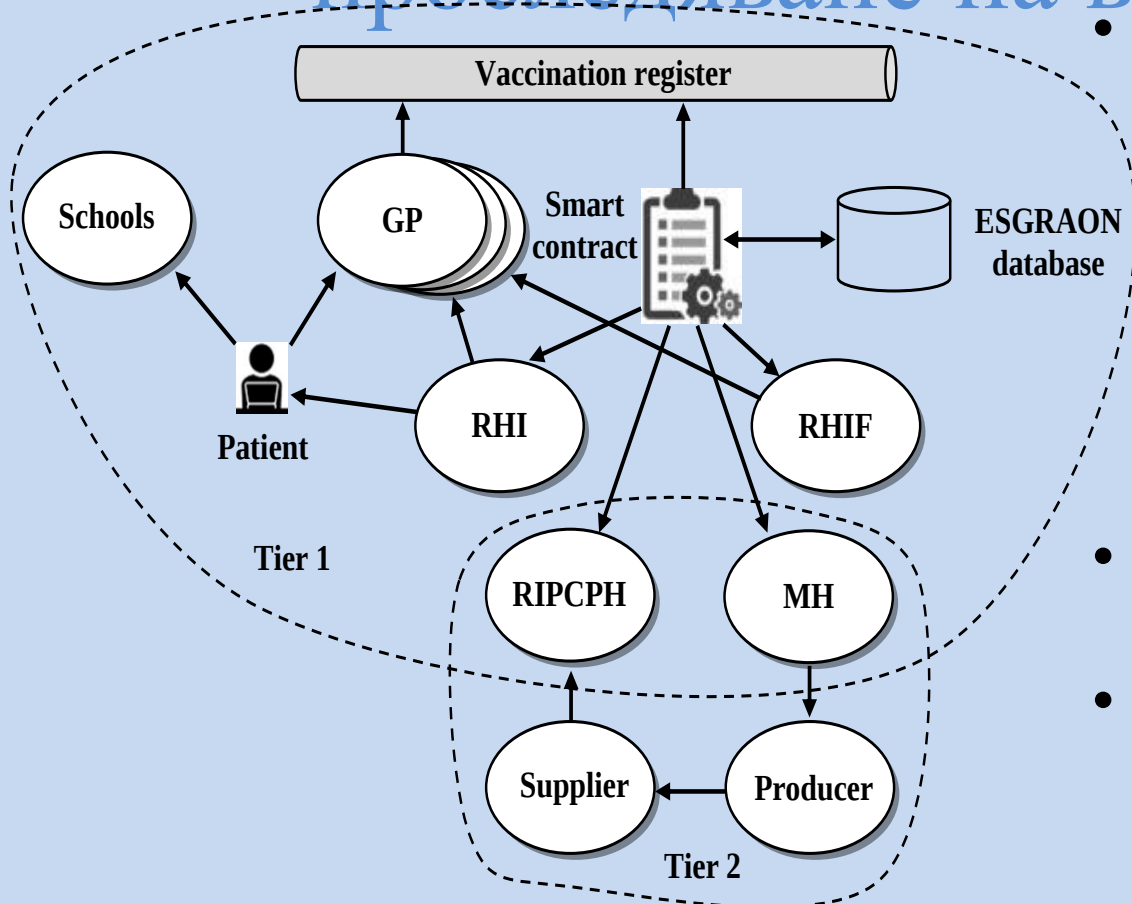
```
NAMES
cli
peer0.org3.example.com
peer0.org2.example.com
peer0.org4.example.com
peer0.org1.example.com
orderer2.example.com
couchdb2
orderer.example.com
couchdb1
couchdb3
orderer3.example.com
orderer4.example.com
orderer5.example.com
couchdb0
```

Блокчейн в здравеопазването – проследяване на ваксинации



- Планиране на ваксинациите за следващата календарна година: 1, 2, 3, 4
- Осигуряване на ваксини: 5, 6, 7
- Извършване и документиране на ваксинацията: 8, 9, 10, 11, 12
- Контрол на ваксинацията: 13, 14, 15

Блокчейн в здравеопазването – проследяване на ваксинации



- Моделът се основава на идеята за създаване на регистър на ваксинациите за всяка задължителна ваксина в имунизационния календар на всеки индивид от населението.
- Този регистър се съхранява в блокчейни.
- Той има достъп до същите институции, както в стандартния модел, но стъпки 1, 3, 9, 10, 11, 12 и 13 са пропуснати.

Публикувано в:

H. Valchanov and V. Aleksieva, "Novel Blockchain - Based Models for Healthcare and Life Science Solution," 2022 International Conference on Communications, Information, Electronic and Energy Systems (CIEES), 2022, pp. 1-5, doi: 10.1109/CIEES55704.2022.9990883.

Блокчейн в здравеопазването – проследяване на ваксинации

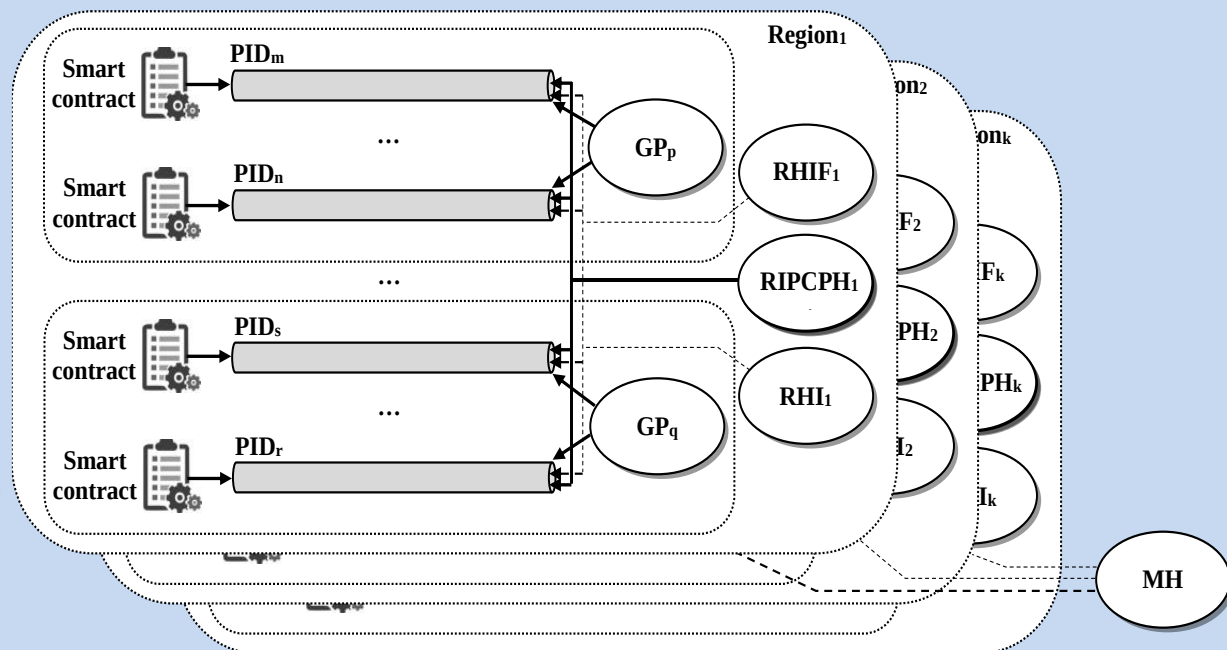
- Предимства
 - Без дублиране на информация;
 - Лесен контрол на имунизирани пациенти от всички колеги;
 - Минимизиран брой неваксинирани пациенти;
 - Точно планиране на броя и вида на ваксините от всяка институция;
 - Значително намаляване на документооборота между институциите;
 - Няма възможност за измама за плащане на неваксинирани ваксинации, за фалшифициране на имунизации и др.;
 - Наличие на информация за съпътстващи реакции към ваксините, което е полезно както за личното здраве при лечението на индивида, така и за обратна връзка с производителя на ваксината.

Публикувано в:

H. Valchanov and V. Aleksieva, "Novel Blockchain - Based Models for Healthcare and Life Science Solution," 2022 International Conference on Communications, Information, Electronic and Energy Systems (CIEES), 2022, pp. 1-5, doi: 10.1109/CIEES55704.2022.9990883.

Проследяване на ваксинации – фокус върху пациента

- За всеки гражданин към момента на раждането му (или имиграцията му) се създава отделен канал, номериран с личния идентификатор (PID) на пациента, който действа като индивидуален имунизационен паспорт.
- Родител, РИОКОЗ, РЗИ, МЗ и РЗОК имат достъп до него чрез смарт договор за четене на информация.
- За регистрация на извършените ваксинации (записване на информация) достъп имат само общопрактикуващите лекари и РИОКОЗ.

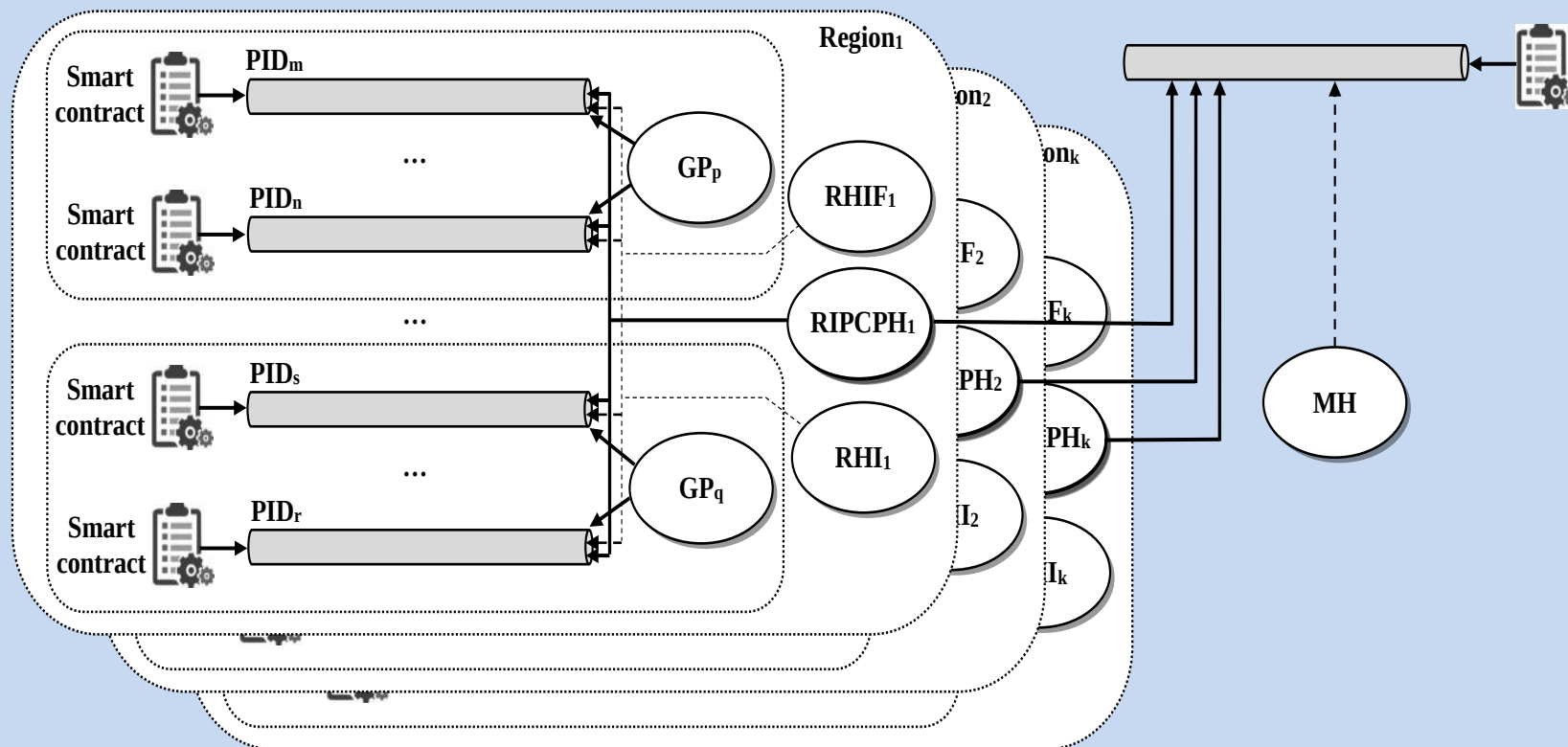


```
type Vaccine struct{
    Date time.Time
    Type string
    VialCode string
    GPcode string
    Reactions string
}
```

Публикувано в:

H. Valchanov and V. Aleksieva, "Novel Blockchain - Based Models for Healthcare and Life Science Solution," 2022 International Conference on Communications, Information, Electronic and Energy Systems (CIEES), 2022, pp. 1-5, doi: 10.1109/CIEES55704.2022.9990883.

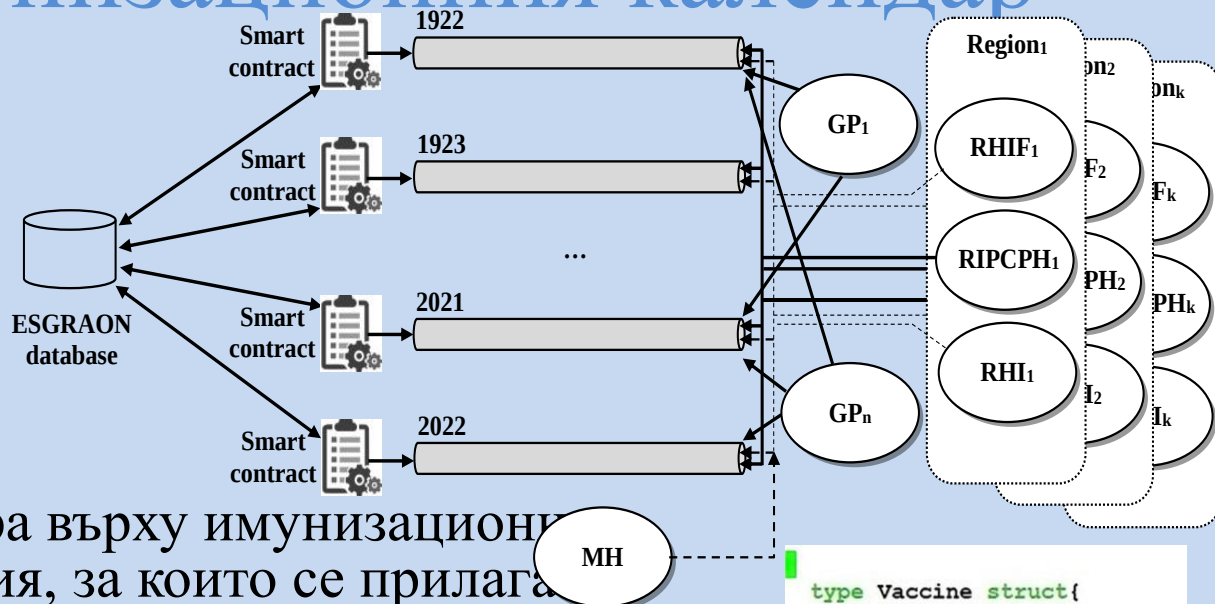
Проследяване на ваксинации – фокус върху пациента



Модифициран модел:

- РИОКОЗ вече въвежда в блока на канала масив с данни за целия ваксиниран период, а МЗ обобщава информацията чрез интелигентен договор на този блокчейн

Проследяване на ваксинации – фокус имунизационния календар



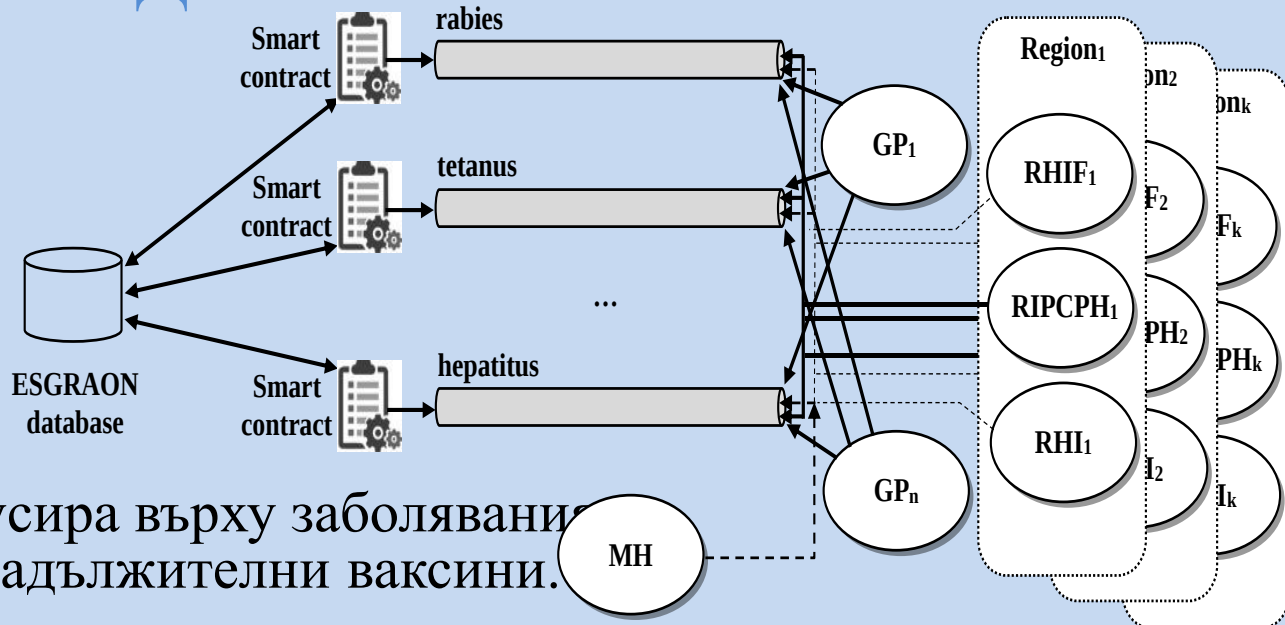
- Този модел се фокусира върху имунизационния календар за заболявания, за които се прилагат задължителни ваксини.
- За всяка година се създава отделен канал. В него се съхранява информация за ваксините за родените през тази година.
- Лекарите имат достъп само до каналите, в които имат пациенти.
- РИОКОЗ, РЗИ, МЗ, РЗОК имат достъп до всички канали.

```

type Vaccine struct{
    Name string
    Use string
    Dose float32
    SerialNumber string
    GPcode string
    Patient Patient
    UsedAt time.Time
    Result string
}

type Patient struct{
    PID string
    FirstName string
    FamilyName string
    Gender string
    Address string
}
    
```

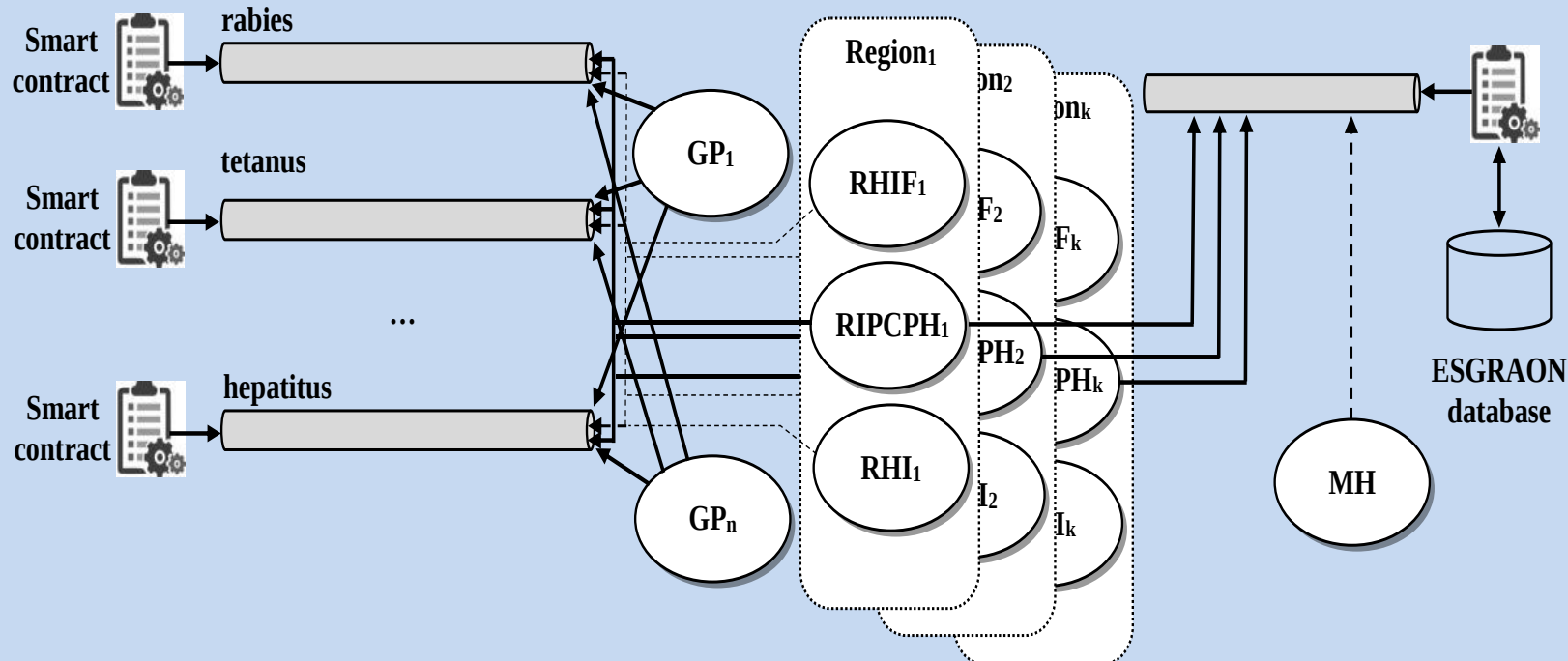
Проследяване на ваксинации – фокус – задължителни ваксини



- Този модел се фокусира върху заболяванията, които се поставят задължителни ваксини.
- Създават се отделни канали за всяко от заболяванията, подлежащи на профилактика чрез задължителна имунизация (в България – 11).
- Записването на направените ваксини се извършва от личните лекари (в България – около 250), веднъж месечно, в един блок.

```
type Vaccine struct{
    PID string
    Date time.Time
    Type string
    VialCode string
    GPcode string
    Reactions string
}
```

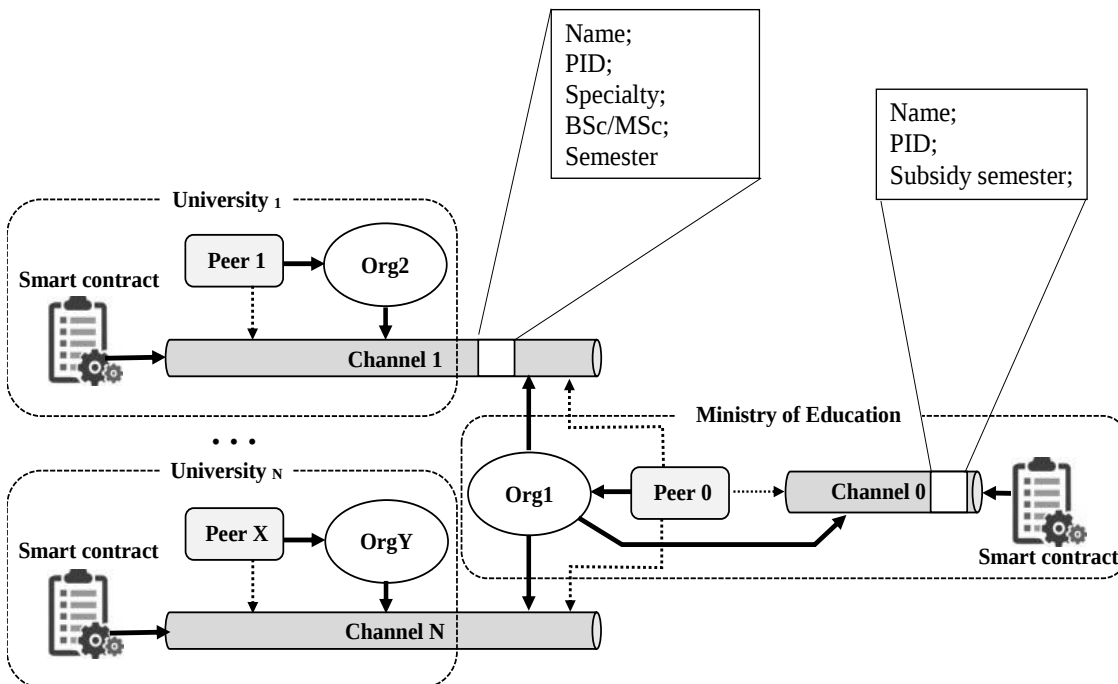
Проследяване на ваксинации – фокус – задължителни ваксини



Модифициран модел:

- Интелигентният договор сравнява данните с информацията от централната система за гражданска регистрация (ЕСГРАОН) за ваксинираните и МЗ получава точна информация за необходимите ваксини.

Блокчейн за проследяване на финансиране в образованието



CONTAINER ID	PORTS	NAMES
cb8ea7d65914		cli
bc0bf9028852	0.0.0.0:7051->7051/tcp	peer0.org1.example.com
09b3fe877196	0.0.0.0:8051->8051/tcp	peer0.org2.example.com
8f9fac3ec528	0.0.0.0:8050->7050/tcp	orderer2.example.com
a8dabbb56610	0.0.0.0:10050->7050/tcp	orderer4.example.com
91c3f9d20416	0.0.0.0:11050->7050/tcp	orderer5.example.com
8f8cba7ac1fa	0.0.0.0:7050->7050/tcp	orderer.example.com
2db2e814f891	4369/tcp, 9100/tcp, 0.0.0.0:5984->5984/tcp	couchdb0
1332918846a7	0.0.0.0:9050->7050/tcp	orderer3.example.com
8ca05dd09eda	4369/tcp, 9100/tcp, 0.0.0.0:6984->5984/tcp	couchdb1

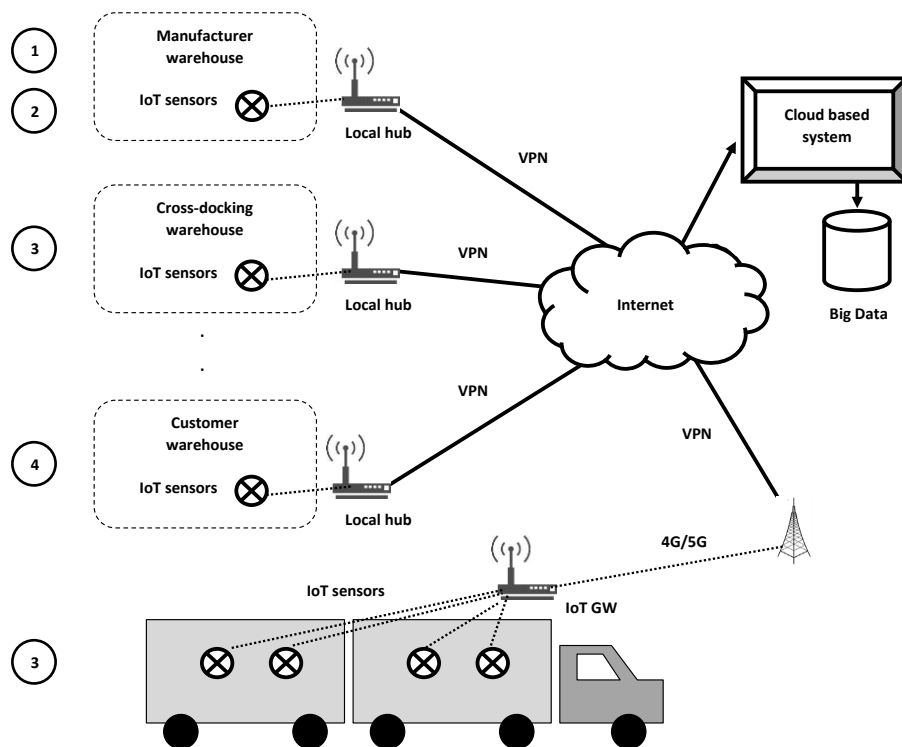
```
2022-08-08 09:45:22.931 UTC [chaincodeCmd] chaincodeInvokeOrQuery ->
DEBU 04b ESCC invoke result: version:1 response:<status:200 > payload:{"n":m250\274\336\226J\3676\036\033\035z\347\273\020k\260\006Nc$367\333^343\224<J\207\317\022\342\001\n\314\001\022\025\n\004lsc022\250\001\032\245\001\n\016student-XV1Bzg\032\222\001\{"Name":\Ivan Dimov Kolev\,\,"PID":\1122334455\,\,"Specialty":\Computer Science\,\,"BM":\Bachelor\,\,"CreatedAt":\0001-01-01T00:00:00Z\,\,"SemesterNum":\2\}\032\003\010\310\001\014\022\005smart\032\0031.0" endorsement:<endorser":\n\007Org1MSP\022\252\006-----BEGIN CERTIFICATE-----nMIICKTCCAc+qAwIBAgIRAKrFFe/Cf+JNMyYkoM39EpAwGqYIgp6n1nVRc2GTCCOkUyKuvTds4MgP+XDC2vF99xCg06ntw=\n-----END CERTIFICATE-----n" signature:"0D\002 >Qd\263\273\224\3159\221\014\224\304-\242\270\255\202\323Qw\225\222\253\Fv\201q\374\250\002 | \234\242A\367\322\314\344\tu\260\224\t\032\376\023\325\300V\253\275j\i\353;-W\252V\235" >
2022-08-08 09:45:22.931 UTC [chaincodeCmd] chaincodeInvokeOrQuery ->
INFO 04c Chaincode invoke successful. result: status:200
```

```
> DEBU 04b ESCC invoke result: version:1 response:<status:200 payload:{"[OK]>>>{\Key":\student-XV1Bzg\,\Record":{\BM":\Bachelor\,\,"CreatedAt":\0001-01-01T00:00:00Z\,\,"Name":\Ivan Dimov Kolev\,\,"PID":\1122334455\,\,"SemesterNum":\2\,\,"Specialty":\Computer Science\}}}} > payload:{"n":023\025\201\256\021\224\263\014\022\005smart\032\0031.0" endorsement:<endorser":\n\007Org1MSP\022\252\006-----BEGIN CERTIFICATE-----nMIICKTCCAc+qAwIBAgIRAKrFFeC6isjEtJG7syHzRjxqlVMwIgp6n1nVRc2GTCCOkUyKuvTds4MgP+XDC2vF99xCg06ntw=\n-----END CERTIFICATE-----n" signature:"0D\002 y\022\321\321k\362\302\014\362\254\206z\234j\005\242q\272s\237\035\353Xh\027\033\224.\220\365\002 V\367y\306p\354\264\326\023\007\001z\351\354\361^375\333F5.\304\247\274\016\301\037tH\014E\016" >
2022-08-08 09:48:36.268 UTC [chaincodeCmd] chaincodeInvokeOrQuery ->
INFO 04c Chaincode invoke successful. result: status:200 payload:{"[OK]>>>{\Key":\student-XV1Bzg\,\Record":{\BM":\Bachelor\,\,"CreatedAt":\0001-01-01T00:00:00Z\,\,"Name":\Ivan Dimov Kolev\,\,"PID":\1122334455\,\,"SemesterNum":\2\,\,"Specialty":\Computer Science\}}}}"
```

Публикувано в:

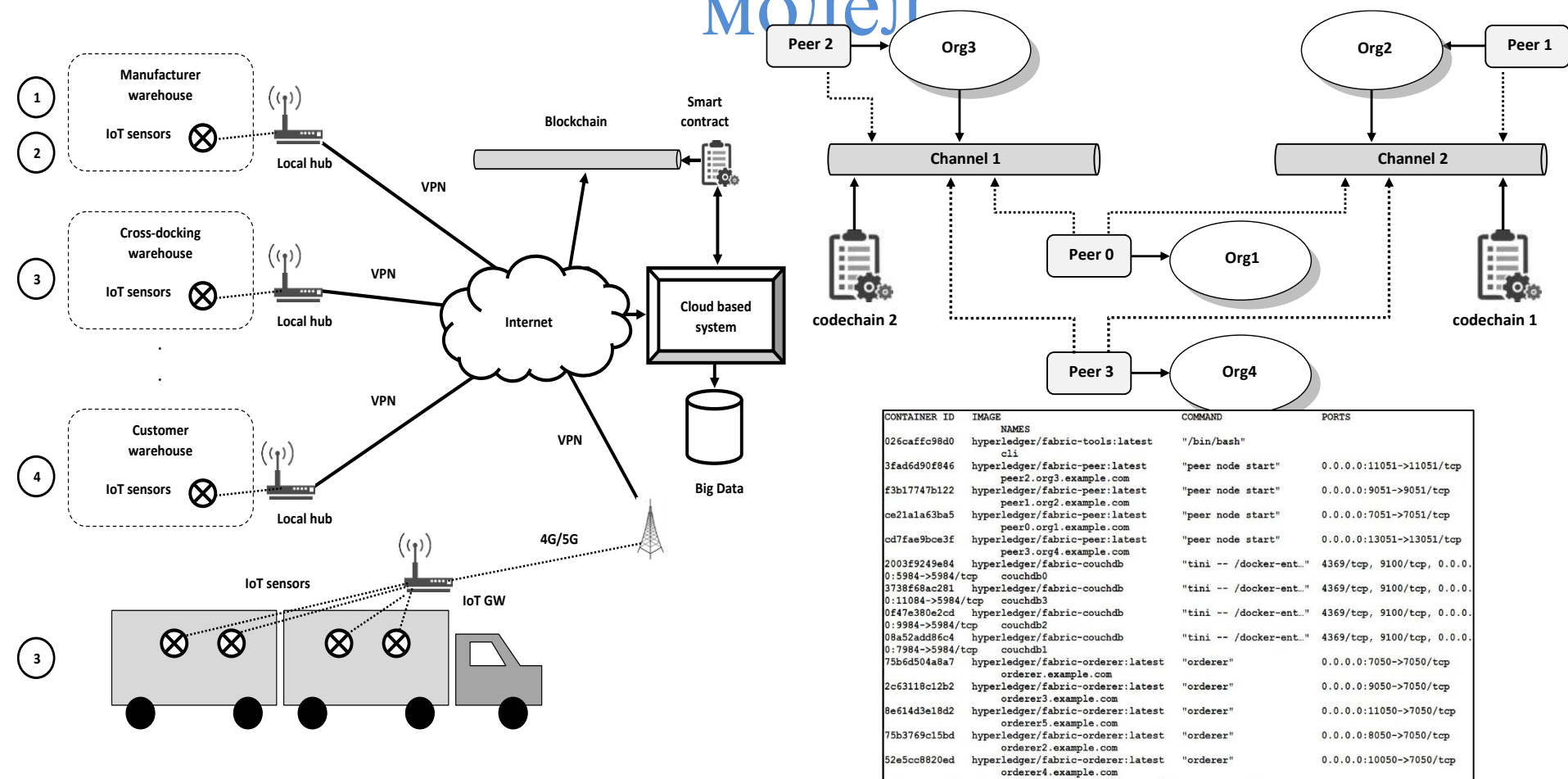
Н. Valchanov and **V. Aleksieva**, "Smart Contracts based on Hyperledger Fabric Blockchain for the Purpose of Higher Education Subsidizing," 2022 International Conference on Communications, Information, Electronic and Energy Systems (CIEES), 2022, pp. 1-5, doi: 10.1109/CIEES55704.2022.9990828.

Приложения в логистиката- транспортиране на стоки- традиционен модел



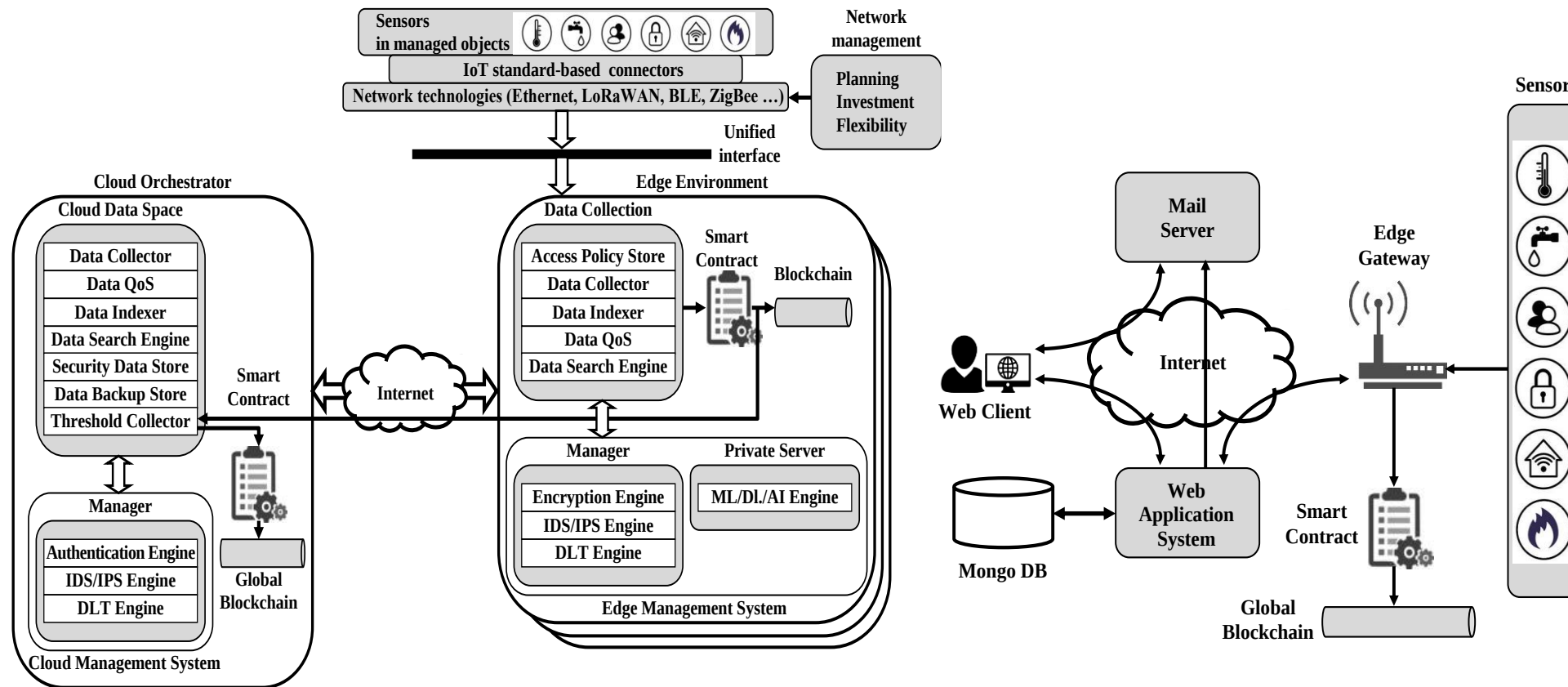
1. Поръчка до доставчик;
2. Заявка за количество в склада на производителя. Проверка на наличността на стоки. Поръчаното количество се премества в транзитен склад/зона за опаковане. Тук стоките могат да бъдат проследени с RFID. Те се подготвят за транспорт чрез инсталиране на IoT сензори. Товарят се на палет/контейнер и оттам в камион;
3. Транспортират се и в зависимост от дестинацията, палетът/контейнерът може да се съхранява в склад за крос-докинг и да се товарят отново на друг транспорт. С IoT сензорите условията за съхранение и транспорт на стоките се контролират в реално време;
4. Доставят се до склада за крайна дестинация. Стоките се разтоварват и разопаковат. При съхранение в склада на получателя информацията от прикрепените IoT сензори вече не е необходима, защото сензорите в неговия склад се наблюдават. И тук стоките могат да бъдат проследени с RFID.

Приложения в логистиката- транспортиране на стоки- блокчейн модел



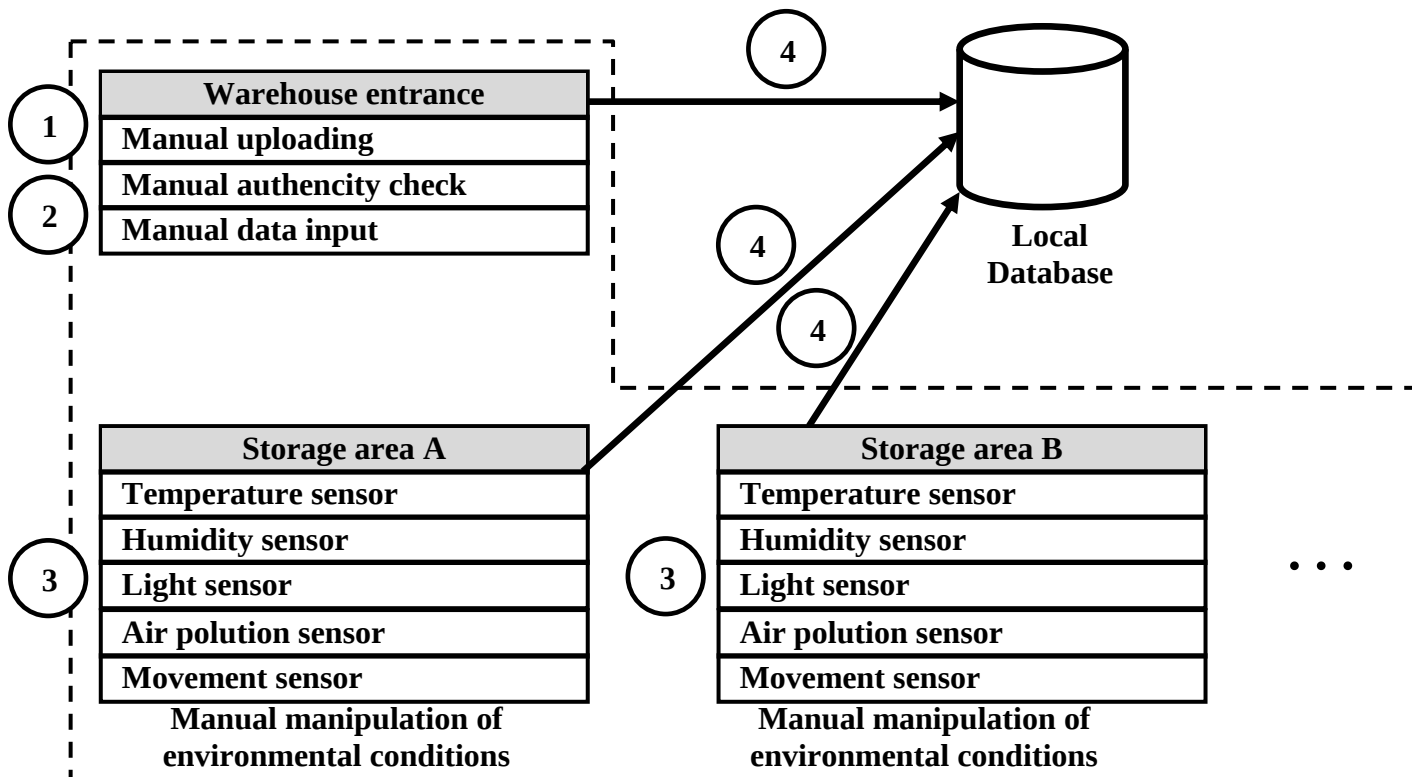
Aleksieva, V.; Valchanov, H.; Haka, A.; Dinev, D. Logistics Model Based on Smart Contracts on Blockchain and IoT. Eng. Proc. 2023, 41, 8.
<https://doi.org/10.3390/engproc2023041008>.

Cloud-Edge IoT-Blockchain модел

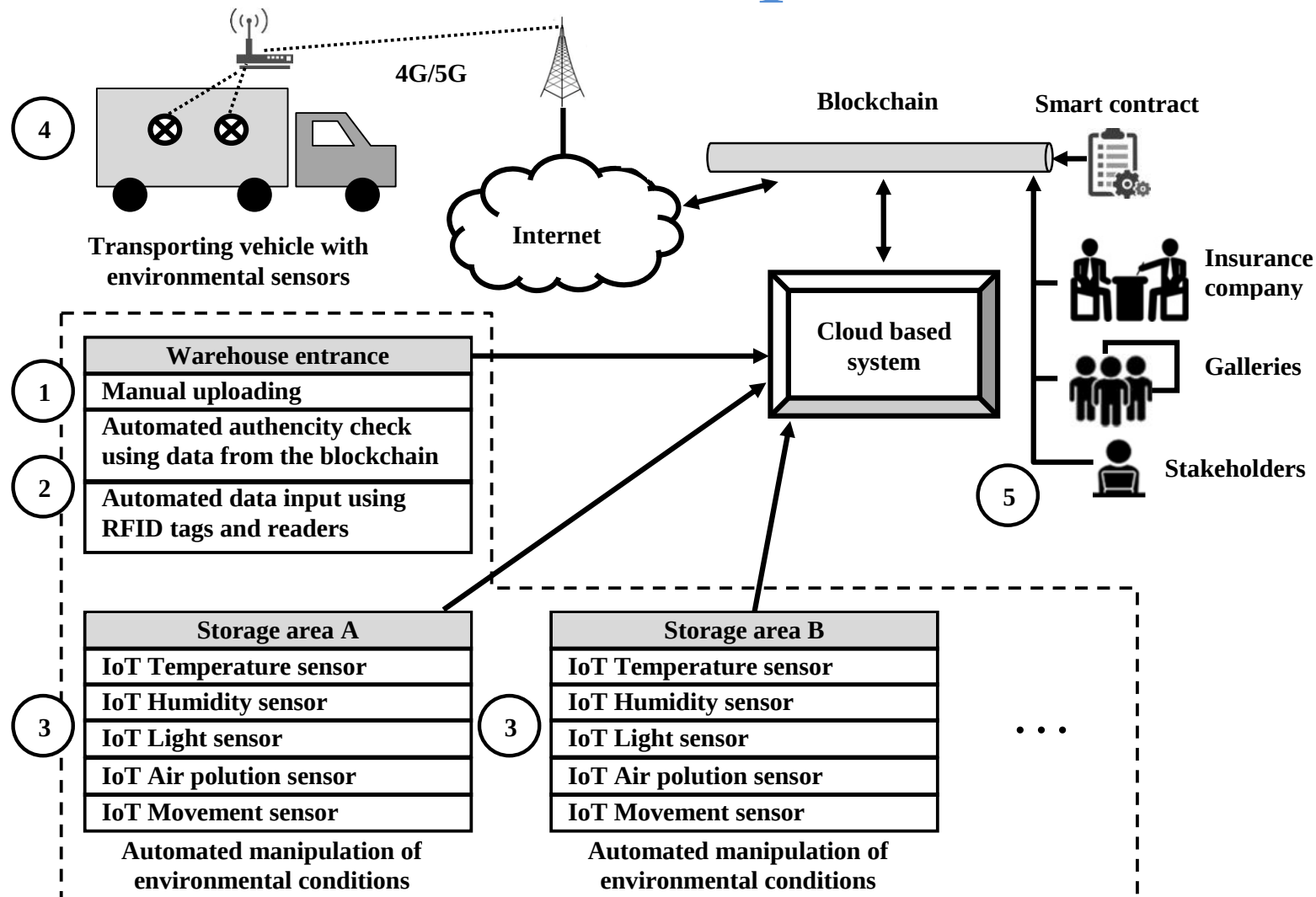


V. Aleksieva, H. Valchanov, A. Haka and D. Dinev, "Model of Controlled Environment based on Blockchain and IoT," *2023 4th International Conference on Communications, Information, Electronic and Energy Systems (CIEES)*, Plovdiv, Bulgaria, 2023, pp. 1-4, doi: 10.1109/CIEES58940.2023.10378795.

Традиционен складов модел



Блокчейн базиран модел



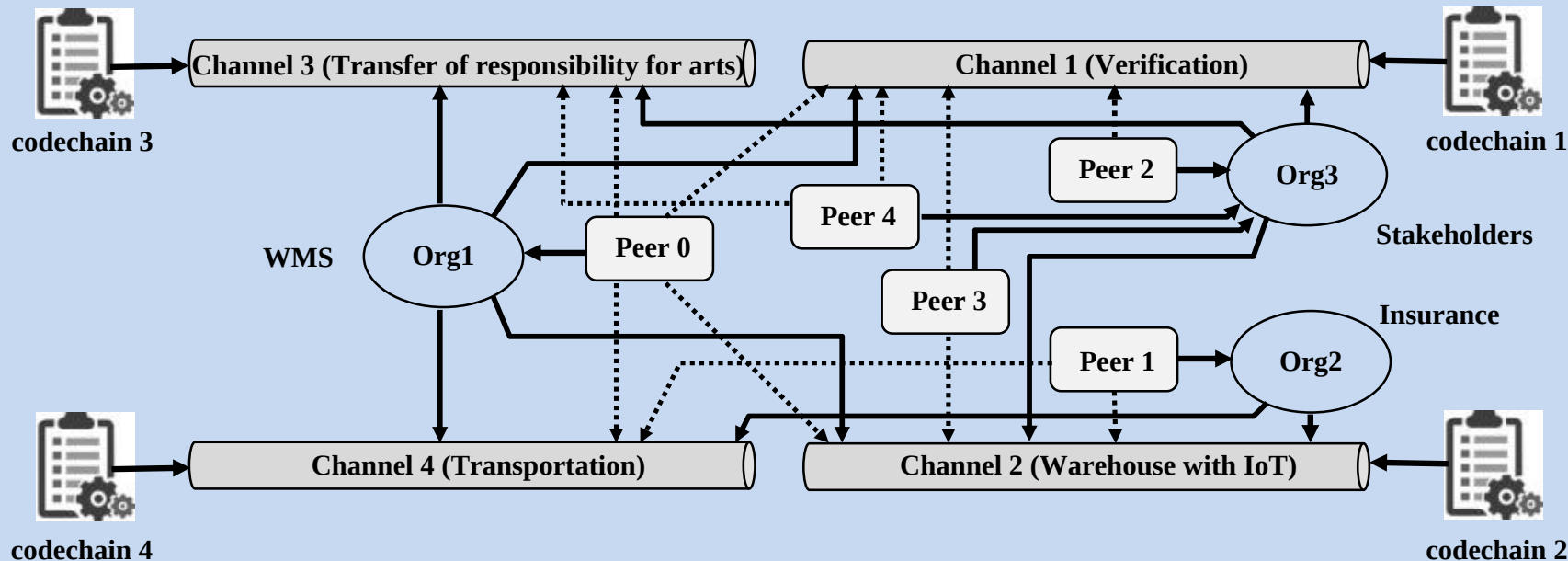
Maleshkov, V.; Valchanov, H.; Aleksieva, V. Blockchain-Based Model for Warehouse Management Systems for Artworks and Collectibles.

Eng. Proc. 2024, 70, 39. EISSN 2673-4591, <https://doi.org/10.3390/engproc2024070039>

9/1/2026

проф. д-р инж. Венета Алексиева

47



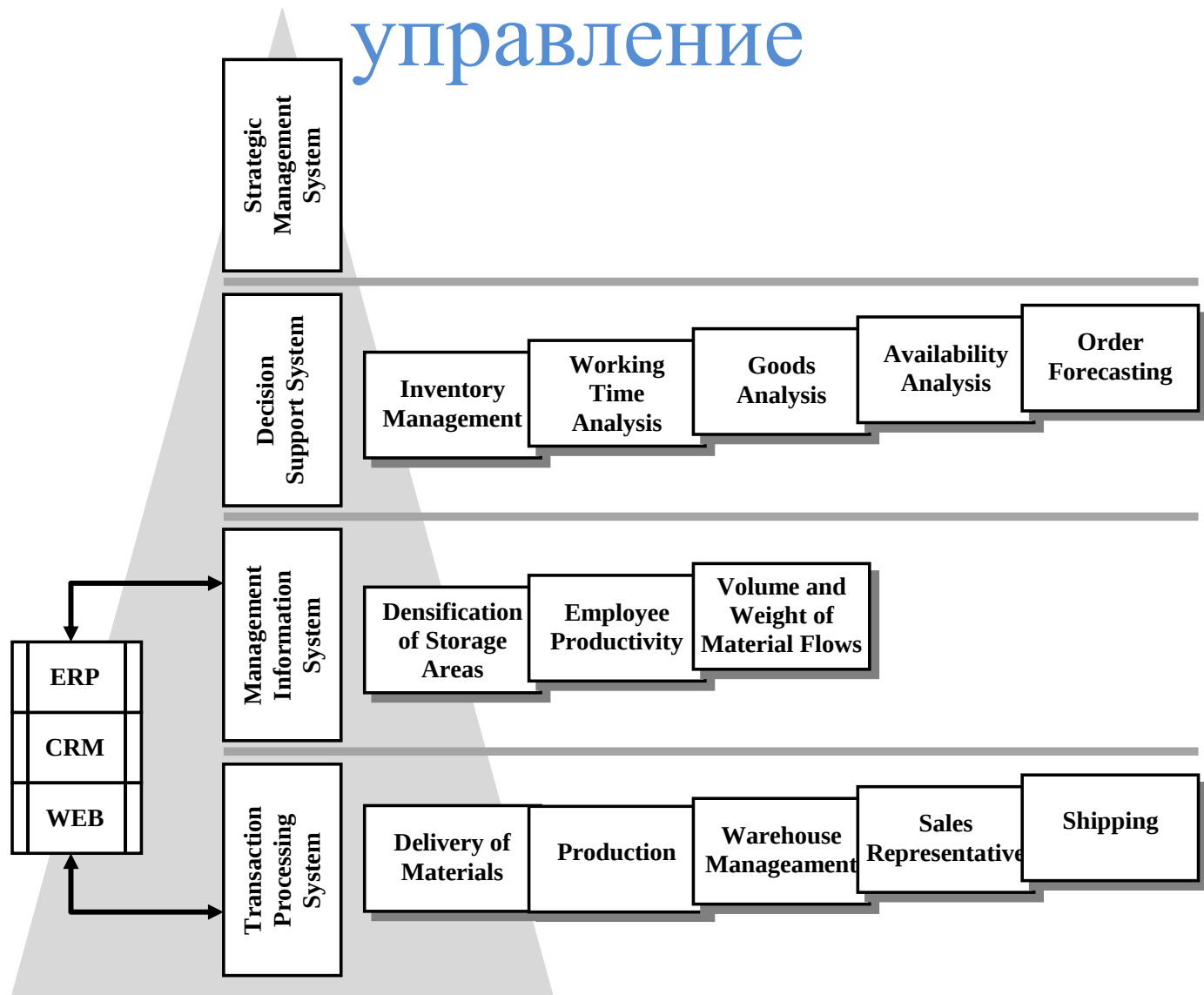
CONTAINER ID	IMAGE	COMMAND	NAMES	CREATED
STATUS	PORTS			
4f1ff47ea88c	hyperledger/fabric-tools:latest	"/bin/bash"		About a mi
Up About a minute			cli	
95dfa44414bd	hyperledger/fabric-peer:latest	"peer node start"		About a mi
Up About a minute	0.0.0.0:11051->11051/tcp		peer0.org3.example.co	
daf742daaf01	hyperledger/fabric-peer:latest	"peer node start"		About a mi
Up About a minute	0.0.0.0:9051->9051/tcp		peer0.org2.example.co	
c141fa9a9442	hyperledger/fabric-peer:latest	"peer node start"		About a mi
Up About a minute	0.0.0.0:7051->7051/tcp		peer0.org1.example.co	
3c5745403fc0	hyperledger/fabric-orderer:latest	"orderer"		About a mi
Up About a minute	0.0.0.0:11050->7050/tcp		orderer5.example.com	
7f7dd81953e4	hyperledger/fabric-couchdb	"tini -- /docker-ent..."		About a mi
Up About a minute	4369/tcp, 9100/tcp, 0.0.0.0:9984->5984/tcp		couchdb2	
808b38da6b40	hyperledger/fabric-orderer:latest	"orderer"		About a mi
Up About a minute	0.0.0.0:9050->7050/tcp		orderer3.example.com	
eadd7b37fdb	hyperledger/fabric-couchdb	"tini -- /docker-ent..."		About a mi
Up About a minute	4369/tcp, 9100/tcp, 0.0.0.0:5984->5984/tcp		couchdb0	
0004f8741c5e	hyperledger/fabric-orderer:latest	"orderer"		About a mi
Up About a minute	0.0.0.0:10050->7050/tcp		orderer4.example.com	
26e7b77a8762	hyperledger/fabric-orderer:latest	"orderer"		About a mi
Up About a minute	0.0.0.0:8050->7050/tcp		orderer2.example.com	
a3279fbc3b7f	hyperledger/fabric-orderer:latest	"orderer"		About a mi
Up About a minute	0.0.0.0:7050->7050/tcp		orderer.example.com	
b4830alee2da	hyperledger/fabric-couchdb	"tini -- /docker-ent..."		About a mi
Up About a minute	4369/tcp, 9100/tcp, 0.0.0.0:7984->5984/tcp		couchdb1	

```

invoke result: version:1 response:<status:200 > payload:"\n o,\2218\301\230\2368\02
2dUQ\177>\035&\355\177\244\250t\027\036K\017:\345\356\205\024\003\320\022\346\001\n
\315\001\022\030\n\004lsc\022\020\n\016\n\010smartcon\022\002\010\003\022\260\001\n
\010smartcon\022\243\001\032\240\001\n\tId-baiCMR\032\222\001{\\"Name\\":\\"David\\",\
\\"TimeStamp\\":\\"2024-04-07 21:19:15\\",\\"Temp\\":\\"23oC/70oF\\",\\"Humidity\\":\\"25.541%
/ 39.2g/m3\\",\\"Light\\":\\"11.85 lux\\",\\"Air\\":\\"0.6%\\",\\"Move\\":\\"No\\"}\032\003\010\
310\001\\\"017\022\010smartcon\032\0031.0\" endorsement:<endorser:\\\"n\007Org1MSP\022\
252\006-----BEGIN CERTIFICATE-----\nMIICKTCCAc+gAwIBAgIRAJ/HN9qfCeyJ9R+cQ7q7hswCgY
IKoZIZj0EAwIwczEL/nMAkGAlUEBhMCVVMxZzARBgNVBAgTCkNhbgG1mb3JuaWExFjAUBgNVBAcTDVNBhbiBG
/ncmFuY21zY28xGTAxBGNVBAoTEG9yZzEuZXhhbXBsZS5jb20xHDAaBgNVBAMTE2Nh/nLm9yZzEuZXhhbXB
sZS5jb20wHhcNMjQwNDExMTAwNTAwHhcnMzQwNDExMTAwNTAw/nWjBqMQswCQYDVQQGEWJVUzETMBEGA1UE
CBMkQ2FsaWZvcz5pYTEwMBQGA1UEBxMN/nU2FuIEZyYW5jaXNjbzENMAAsGA1UECzMEcGVlcmB0GA1UEA
uMWMcGVlcmB0GA1UEBxMN/nMS5leGFTcGx1LnNvbTBZMBMGByqGSM49AgEGCCqGSM49AwEHA0IABNRci+3cky5L/n
uXtsdB3UzP0sEXJYn9yT5jUz6gxX91WVdoKIOzG+8Akjm/swKKsAkxzZwcnKZCTC\nnNrRrvjPxdPajjTTBLM
A4GA1UdDwEB/wQEAwIHGDAMBgNVHRMBAf8EAjAAMCsGA1Ud\nIwQkMCKAIKkDOBBE2pvAWLVYBhNGDsb/
/m0h1OFMt2bsQ1ZuMqMaGCCqGSM49\nnBAMCA0gAMEUCIQCBMeAcXrDaXtTT5r6QUki8xtgBvOocN31MgV
psumpuQIgMgLUNGiTiJ51kI9GhcvfsNXWD5+4B/Zw7niqFj19ea8s=\\n-----END CERTIFICATE-----\
\\n\" signature:\"0E\002!\000\037\264\243E\334\212\362\263\366\346\223\236\345\337\30
1L\354\242\347\364\014\007\022 \221qmb\207\337\002 >i\301\177~+\210\033Fwn\230\352?
\374K\361\2319P7\374\034\331\237\246\253\371V\270\237\331\" >
2024-04-11 10:19:04.943 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 04c Chain
code invoke successful. result: status:200

```

Иерархичен модел на логистично управление



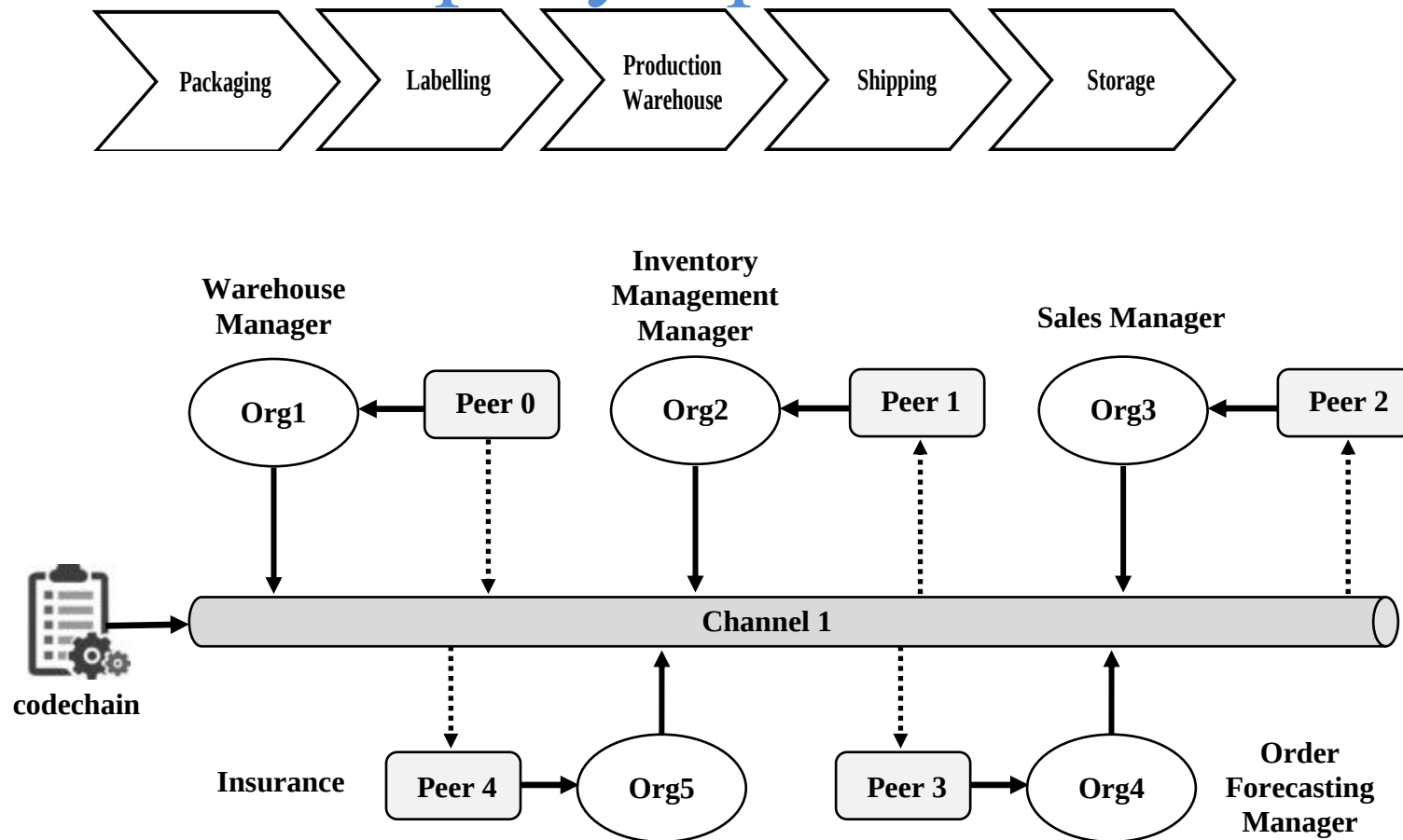
Aleksieva, V.; Valchanov, H.; Maleshkov, V.; Haka, A. Blockchain Solutions for Logistic Management. Blockchains 2024, 2, 445-457.
<https://doi.org/10.3390/blockchains2040019>

9/1/2026

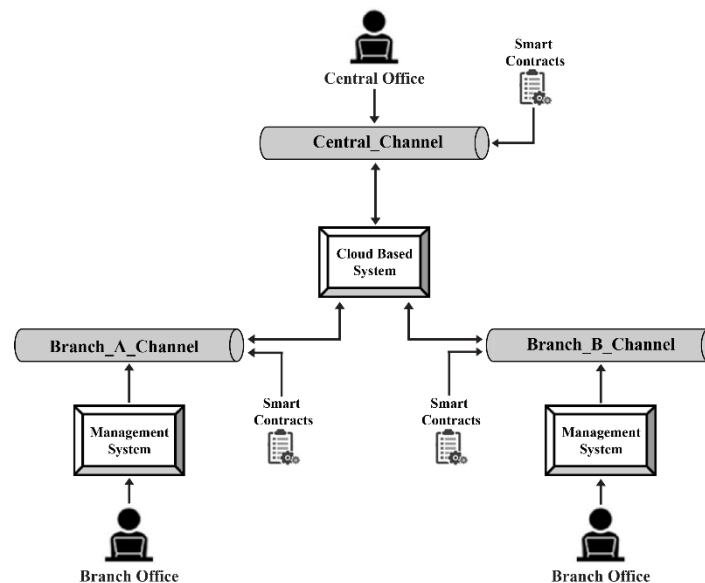
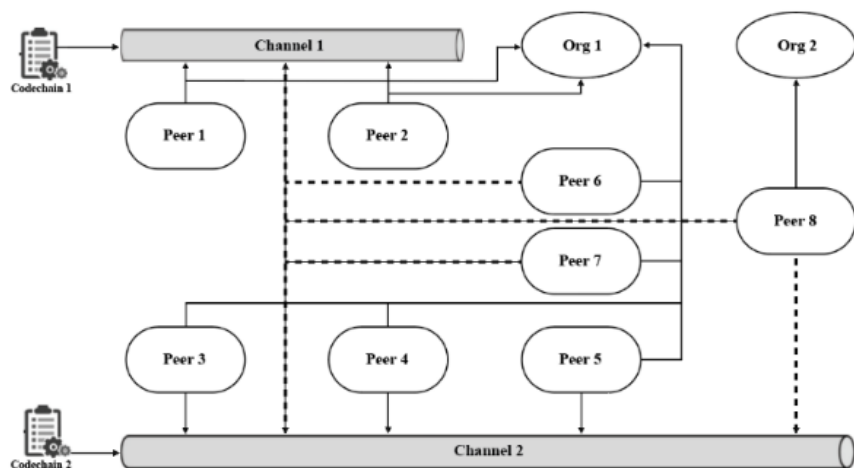
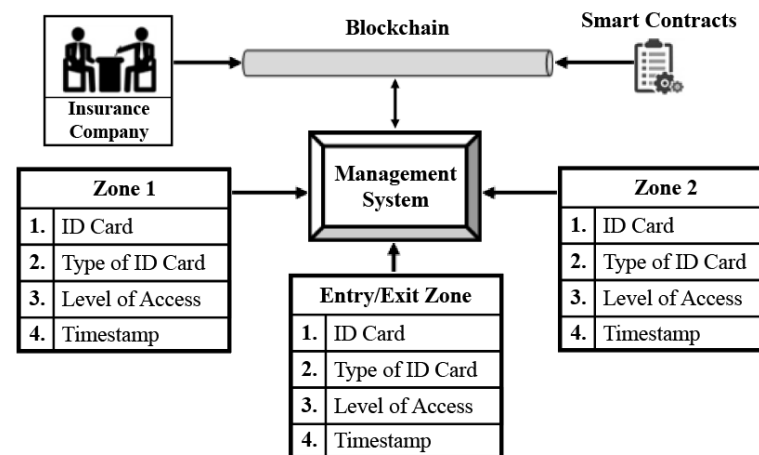
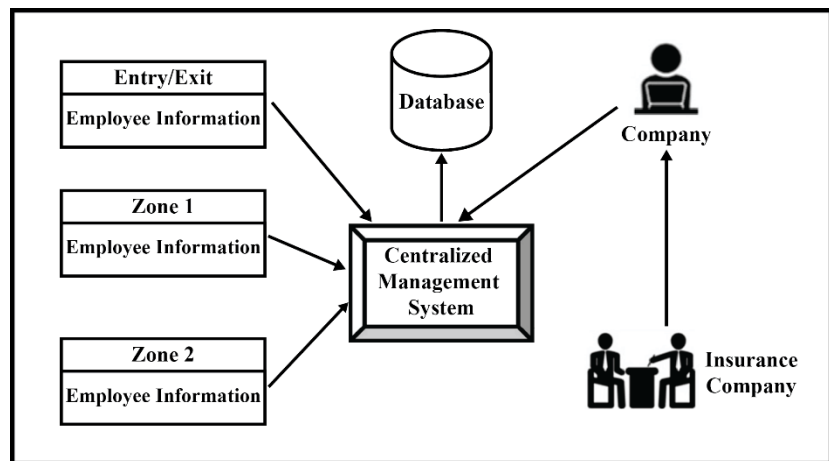
проф. д-р инж. Венета Алексиева

49

Доставка на продукция до дистрибуторски склад

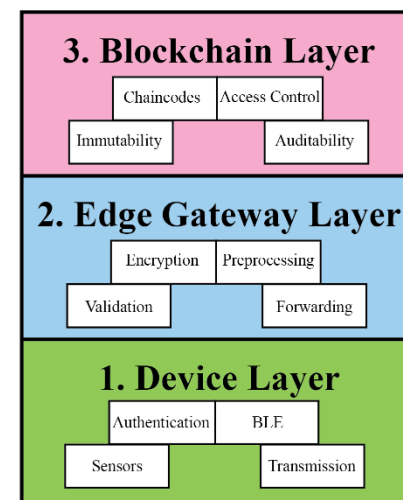
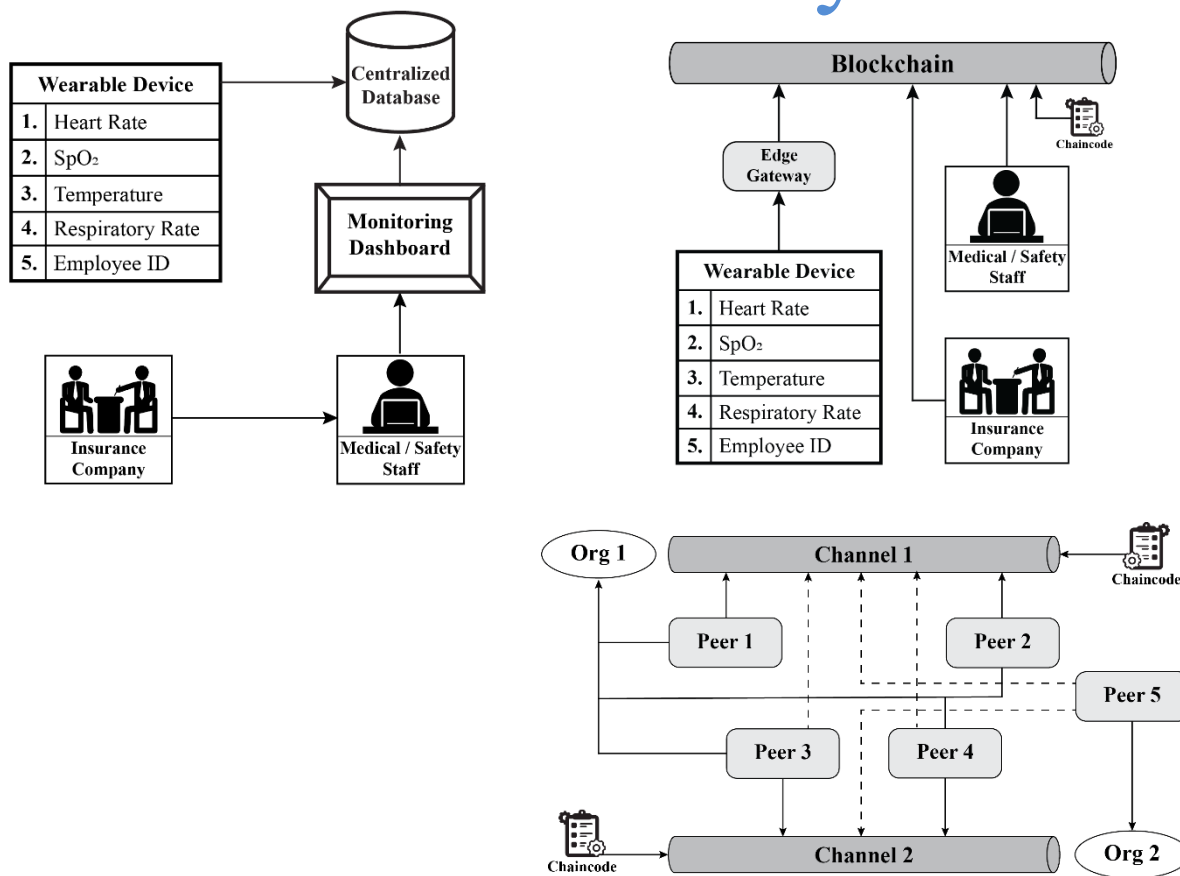


Проследяване на служители



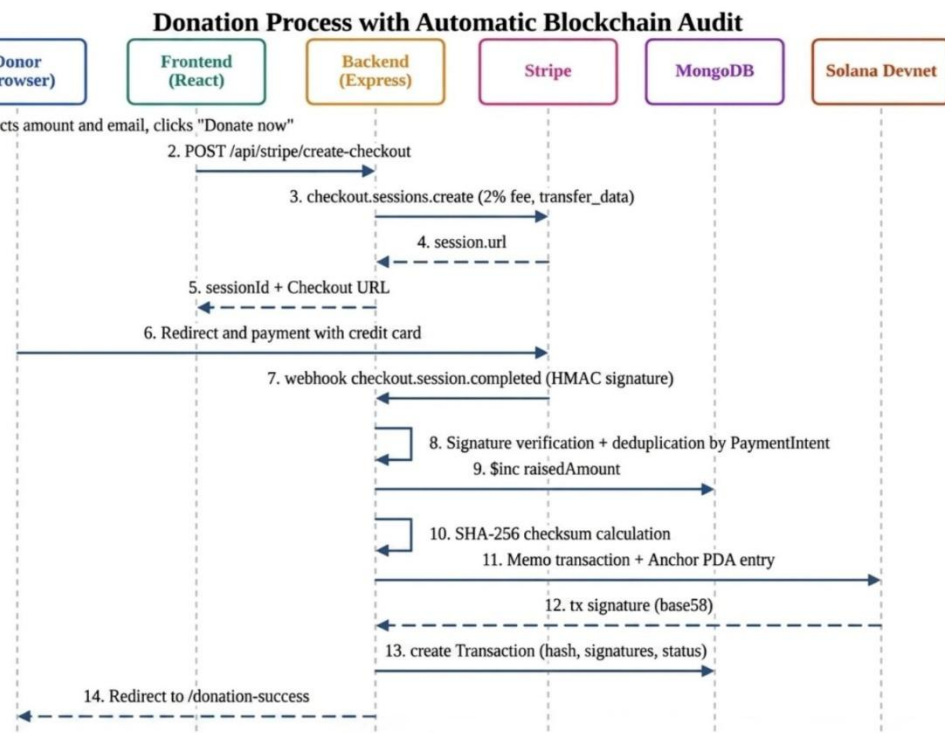
V. Maleshkov, V. Aleksieva and H. Valchanov, "Blockchain model for tracking employees' location in the company's premises " 2025, EEPES, Bulgaria, 2025, pp. 1-14.

Блокчейн приложения в логистиката за следене на жизнени показатели на служители



V. Maleshkov, V. Aleksieva and H. Valchanov, "Blockchain-Based Model for Monitoring Employees Vital Signs," 2025 6th International Conference on Communications, Information, Electronic and Energy Systems (CIEES), Ruse, Bulgaria, 2025, pp. 1-6, doi: 10.1109/CIEES66347.2025.11300147.

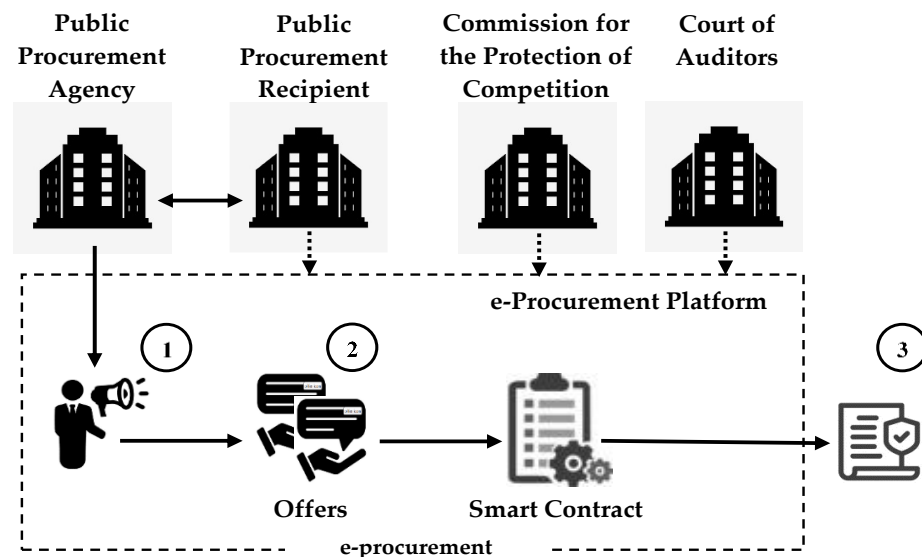
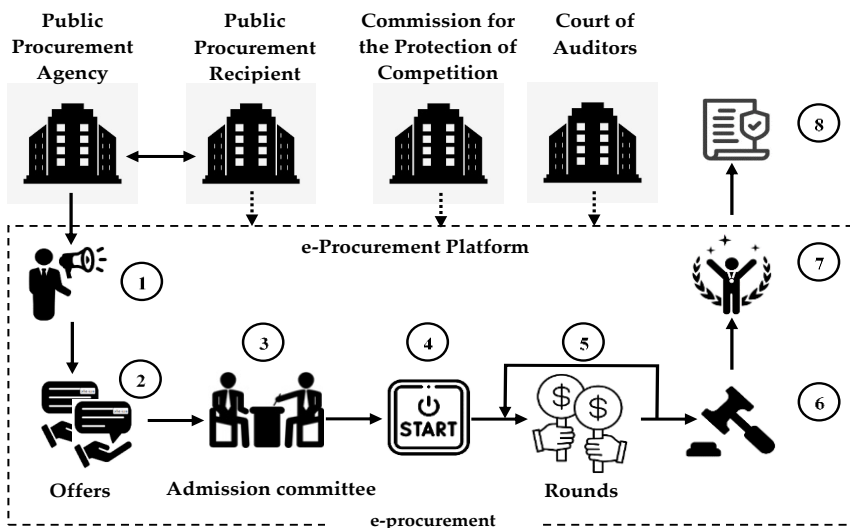
Блокчейн за дарения



Публикувано в:

D. Mehmed, **V. Aleksieva** and H. Valchanov, "Web-based donation system with QR codes and blockchain," 2026 International Conference on Communications, Information, Electronic and Energy Systems (CIEES), 2022, pp. 1-7,

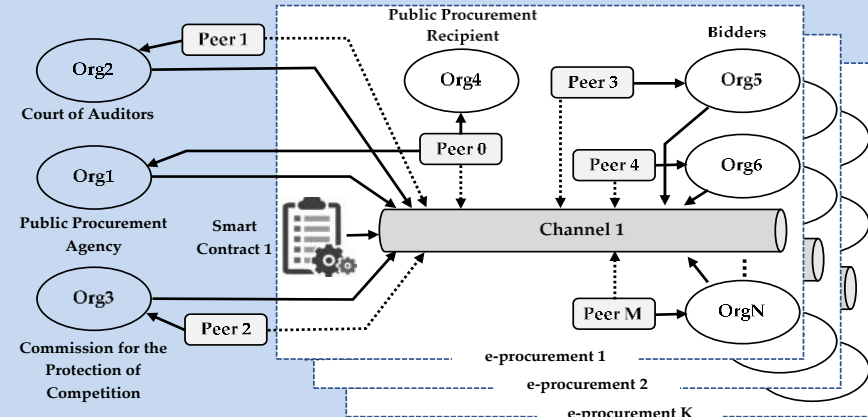
Блокчейн за е-търгове



Публикувано в:

V. Aleksieva and H. Valchanov, "Blockchain solutions for e-tenders in public procurements," 2026, blockchains, mdpi, pp.1-13.

Блокчейн за е-търгове



CONTAINER ID	IMAGE	NAMES	COMMAND	CREATED	STATUS	PORTS
ba77e8520515	hyperledger/fabric-tools:latest	cli	"/bin/bash"	20 minutes ago	Up 19 minutes	
4d9208e8c227	hyperledger/fabric-peer:latest	cli	"peer node start"	20 minutes ago	Up 20 minutes	0.0.0.0:7051->
7051/tcp	hyperledger/fabric-peer:latest	peer0.org1.example.com	"tini -- /docker-ent..."	20 minutes ago	Up 20 minutes	4369/tcp, 5984
54935553917b	hyperledger/fabric-couchdb	couchdb4	"orderer"	20 minutes ago	Up 20 minutes	0.0.0.0:10050->
/tcp, 9100/tcp, 0.0.0.0:13984->13984/tcp	hyperledger/fabric-orderer:latest	orderer4.example.com	"orderer"	20 minutes ago	Up 20 minutes	0.0.0.0:8050->
51367e542d0a	hyperledger/fabric-orderer:latest	orderer2.example.com	"orderer"	20 minutes ago	Up 20 minutes	0.0.0.0:7050->
>7050/tcp	hyperledger/fabric-orderer:latest	orderer.example.com	"tini -- /docker-ent..."	20 minutes ago	Up 20 minutes	4369/tcp, 5984
fcaaf63b7c26	hyperledger/fabric-couchdb	couchdb2	"orderer"	20 minutes ago	Up 20 minutes	0.0.0.0:9050->
7050/tcp	hyperledger/fabric-couchdb	couchdb0	"tini -- /docker-ent..."	20 minutes ago	Up 20 minutes	4369/tcp, 9100
6b5ac546a1aa	hyperledger/fabric-couchdb	couchdb1	"tini -- /docker-ent..."	20 minutes ago	Up 20 minutes	4369/tcp, 5984
7050/tcp	hyperledger/fabric-couchdb	couchdb3	"tini -- /docker-ent..."	20 minutes ago	Up 20 minutes	4369/tcp, 5984
42ee1671567	hyperledger/fabric-couchdb	couchdb5	"tini -- /docker-ent..."	20 minutes ago	Up 20 minutes	4369/tcp, 5984
/tcp, 9100/tcp, 0.0.0.0:9984->9984/tcp	hyperledger/fabric-orderer:latest	orderer3.example.com	"tini -- /docker-ent..."	20 minutes ago	Up 20 minutes	4369/tcp, 5984
5538629f624a	hyperledger/fabric-orderer:latest	orderer5.example.com	"tini -- /docker-ent..."	20 minutes ago	Up 20 minutes	4369/tcp, 5984
7050/tcp	hyperledger/fabric-orderer:latest	orderer	"tini -- /docker-ent..."	20 minutes ago	Up 20 minutes	4369/tcp, 5984
9424c7f5d01b	hyperledger/fabric-couchdb	couchdb0	"tini -- /docker-ent..."	20 minutes ago	Up 20 minutes	4369/tcp, 5984
/tcp, 0.0.0.0:5984->5984/tcp	hyperledger/fabric-orderer:latest	orderer	"tini -- /docker-ent..."	20 minutes ago	Up 20 minutes	4369/tcp, 5984
569e3f7b89b	hyperledger/fabric-orderer:latest	orderer	"tini -- /docker-ent..."	20 minutes ago	Up 20 minutes	4369/tcp, 5984
>7050/tcp	hyperledger/fabric-couchdb	couchdb1	"tini -- /docker-ent..."	20 minutes ago	Up 20 minutes	4369/tcp, 5984
0dec31613bb6	hyperledger/fabric-couchdb	couchdb2	"tini -- /docker-ent..."	20 minutes ago	Up 20 minutes	4369/tcp, 5984
/tcp, 9100/tcp, 0.0.0.0:7984->7984/tcp	hyperledger/fabric-couchdb	couchdb3	"tini -- /docker-ent..."	20 minutes ago	Up 20 minutes	4369/tcp, 5984
f432ad14ee3b	hyperledger/fabric-couchdb	couchdb4	"tini -- /docker-ent..."	20 minutes ago	Up 20 minutes	4369/tcp, 5984
/tcp, 9100/tcp, 0.0.0.0:11984->11984/tcp	hyperledger/fabric-couchdb	couchdb5	"tini -- /docker-ent..."	20 minutes ago	Up 20 minutes	4369/tcp, 5984
9bca84b085f	hyperledger/fabric-couchdb	couchdb6	"tini -- /docker-ent..."	20 minutes ago	Up 20 minutes	4369/tcp, 5984
/tcp, 9100/tcp, 0.0.0.0:15984->15984/tcp						

```
2026-04-11 11:15:06.873 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> DEBU 04b ESCC i
nvoke result: version: 1 response: <status:200 > payload: {"Procurement": "Highway co
nstruction Lot 2", "TimeStamp": "2026-04-11 11:15:05.452", "Bidder": "Compass Ltd", "Bid"
: "1000", "smartcontract endorsement: <endorser: "OrgMSP">
2026-04-11 11:15:06.873 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 04c chainc
ode invoke successful. result: status:200
```

```
PROCEDURE Start_Auction (Participants_List, Award_Criteria, Auction_Params):
```

```
// FILTERING CANDIDATES APPROVED FOR TECHNICAL ASSESSMENT
Approved_Participants = Admitted (Participants_List)
```

```
// SET STARTING VALUES FROM INITIAL BIDS
```

```
FOR EVERY Participant IN Approved_Participants:
    Curr_Offer [Participant] = Participant.Initial_Offer
    IF Award_Criteria == "Optimal_Ratio":
        Tech_Points [Participant] = Participant.Tech_Assessment
```

```
Curr_Round = 1
Active_Auction = TRUE
Rounds_No_Change = 0
```

```
// MAIN BIDDING LOOP
```

```
WHILE Active_Auction == TRUE:
    Bids_Submitted_In_Round = 0
    Best_Offer_Before_Round = Get_Best_Offer (Curr_Offer)
```

```
FOR EVERY Participant IN Approved_Participants:
```

```
// CHECKING FOR PARTICIPANT ACTIVITY IN THIS ROUND
IF Participant.Submit_New_Offer (Curr_Round):
    New_Offer = Participant.Enter_Offer (Curr_Round)
```

```
// CHECK IF THE OFFER MEETS THE BIDDING STEP
```

```
IF Validate_Step (New_Offer, Curr_Offer [Participant], Auction_Params.Step):
    Curr_Offer [Participant] = New_Offer
    Bids_Submitted_In_Round = Bids_Submitted_In_Round + 1
ELSE:
    Reject_Offer (Participant, "Invalid Step")
```

```
ELSE:
    // THE LAST VALID OFFER IS RETAINED
    Curr_Offer [Participant] = Curr_Offer [Participant]
```

```
// AUTOMATIC RECALCULATION AND TIME RANKING AFTER THE ROUND
```

```
FOR EVERY Participant IN Approved_Participants:
```

```
IF Award_Criteria == "Lowest_Price":
    Final_Points [Participant] = Curr_Offer [Participant].Price
```

```
ELSE IF Award_Criteria == "Optimal_Ratio":
```

```
// MATHEMATICAL FORMULA FOR COMPLEX ASSESSMENT
```

```
Final_Points [Participant] =
    Recalculate_Comprehensive_Assessment (Curr_Offer [Participant].Price,
    Tech_Points [Participant])
```

```
Temporary_Ranking = Sort_Participants (Approved_Participants, Final_Points, Award_Criteria)
Show_Results (Temporary_Ranking)
```

```
// CHECKING THE END OF AUCTION CONDITIONS
```

```
Best_Offer_After_Round = Get_Best_Offer (Curr_Offer)
```

```
IF Best_Offer_After_Round == Best_Offer_Before_Round:
```

```
    Rounds_No_Change = Rounds_No_Change + 1
```

```
ELSE:
    Rounds_No_Change = 0
```

```
IF Curr_Round >= Auction_Params.Max_Number_Rounds:
```

```
    Active_Auction = FALSE
```

```
ELSE IF Rounds_No_Change >= Auction_Params.Max_Number_Rounds.No_Activity:
```

```
    Active_Auction = FALSE
```

```
ELSE IF Curr_Time () >= Auction_Params.End_Time:
```

```
    Active_Auction = FALSE
```

```
ELSE:
    Curr_Round = Curr_Round + 1
```

```
// FINAL RANKING AND EXPORT
```

```
Final_Ranking = Sort_Participants (Approved_Participants, Final_Points, Award_Criteria)
```

```
Export_To_Bid_Evaluation (Final_Ranking)
```

```
RETURN Final_Ranking
```

Публикувано в:

V. Aleksieva and H. Valchanov, "Blockchain solutions for e-tenders in public procurements," 2026, blockchains, mdpi, pp.1-13.

Блокчейн приложения за гласуване в национални избори

- Сигурно удостоверяване на самоличността на избирателите, намалявайки риска от фалшиви гласове.
- Записване на гласовете по непроменим и защитен от подправяне начин, гарантирайки целостта на процеса на гласуване.
- Подобряване на прозрачността и одитируемостта на целия процес на гласуване, включително преброяването на гласовете.
- Сигурно и проверимо дистанционно гласуване, т.е. даване на възможност на хората да гласуват от отдалечени места.
- Проверка с умен договор на самоличността на лицата, участващи в дейности по гласуване и управление, намалявайки риска от измами с самоличност.

Блокчейн приложения за гласуване на акционери в корпорации

- Гласуване чрез пълномощник за акционери в корпорации или членове на организации, за да гласуват сигурно, намалявайки нуждата от посредници.
- Създаване на децентрализирани самоуправляващи се организации, които вземат решения чрез интелигентни договори и механизми за гласуване, кодирани в блокчейна.
- Проверка с умен договор на самоличността на лицата, участващи в дейности по гласуване и управление, намалявайки риска от измами с самоличност.
- Гласуване на базата на токени, т.е. позволяване на заинтересованите страни да гласуват пропорционално на своите притежания (токени), както се наблюдава в някои модели на управление, базирани на блокчейн.
- Извършване на непроменими одити на процесите на гласуване и управление, осигуряване на съответствие и прозрачност.

Блокчейн приложения за цифрова интелектуална собственост

- Блокчейн може да се използва за установяване и проверка на собствеността и произхода на цифрови активи, като например творчески произведения, патенти, търговски марки и цифрови медии.
- Създателите на съдържание могат да поставят времеви печат на работата си в блокчейн, предоставяйки неоспорими доказателства за датата на създаване. Това може да се използва за установяване на авторски права и доказване на собственост в случай на спорове.
- Интелигентните договори могат да автоматизират плащанията на авторски права, лицензионни споразумения и други договорни задължения.
- Интелигентните договори могат да наложат условията на лицензионните споразумения, като гарантират, че авторските възнаграждения се разпределят автоматично и справедливо между създателите и притежателите на права въз основа на предварително дефинирани условия и данни за употреба.

Блокчейн приложения за материална интелектуална собственост

- За физически продукти, блокчейн може да се използва за създаване на цифрови сертификати за автентичност. Това помага за предотвратяване на фалшифицирани стоки и гарантира, че потребителите купуват оригинални продукти със свързаните с тях права върху интелектуална собственост.
- В индустрии като модата и луксозните стоки, блокчейн помага за гарантиране на автентичността на продуктите и помага за идентифициране и справяне с нарушения на интелектуална собственост.

Блокчейн приложения за патенти

- Управлението на патентната информация, включително подаване, съдебно преследване и поддръжка, може да бъде по-ефективно с помощта на блокчейн технологията.
- Това може да помогне на изобретателите и организациите да поддържат сигурен и прозрачен регистър на своите патенти.
- Блокчейн може да улесни децентрализираните пазари за покупка и продажба на права върху интелектуална собственост, което улеснява създателите и компаниите да монетизират своята интелектуална собственост директно с други, без да е необходимо участието на посредници.

Блокчейн и Виртуалния свят

- Европейската комисия (ЕК) през 2022г. предостави конкретно определение за метавселена (виртуален свят): „Виртуалните светове са устойчиви, завладяващи среди, базирани на технологии, включително 3D и разширена реалност (XR), които правят възможно смесването на физически и цифрови светове в реално време за различни цели, като проектиране, създаване на симулации, сътрудничество, учене, социализиране, извършване на транзакции или предоставяне на забавление“
- Blockchain Observatory and Forum през 2022г. Определи : „ Метавселената е продукт на технологично обусловена промяна с обобщено въздействие чрез устойчиви и адаптивни цифрови преживявания.“
- Изследването на виртуалните светове води до две различни визии:
 - затворен виртуален свят – контролирани от мощни организации в рамките на собствени домейни
 - отворен виртуален свят - оперативно съвместими, отворени системи, подобрени от съвременни цифрови приложения, възможност за универсално споделени регистри на цифрови и физически активи (NFT или блокчейн).
- Отворените блокчейни без разрешения са от решаващо значение за оформянето на отворените виртуални светове, предлагайки създаване на гъвкави системи.

Блокчейн, Web 3.0, AI и IoT

- Блокчейните предлагат гъвкавостта за интегриране с настоящите Web 3.0 приложения
- Днес или утре ще се появят саморегулиращи се градове или райони със специфични политики, правила и критерии за достъп.
- Градове с умна инфраструктура, енергийна ефективност и сигурност, управлявани от умни договори в блокчейн, с ИИ и IoT.



Въпроси ?

Благодаря за вниманието !