Session I : Klaytn Architecture

김정현 / Ground X

Account and Transaction Model in Klaytn

김정현, Colin

Lead of Platform & SDK, Ground X

- Account model
- Transaction model
- Improving dev/test environment

Software Engineer, Samsung Electronics

- Improving Tizen development environment
- Developing AI software stack for mobile

Ph.D. in Computer Science and Engineering

- System software
- Computer architecture
- Parallel programming model
- GPGPU

TABLE OF CONTENTS

- ·What are accounts and transactions?
- ·Usability considerations
  - User’s perspective
  - Service provider’s perspective
  - Platform developer’s perspective
- ·Account model
- ·Transaction model
- ·Conclusion

Account and Transaction

- Account
  - A data structure storing information of users and contracts
    - Nonce
    - Balance
    - CodeHash
    - StorageRoot

Account and Transaction

- Account
  - A data structure storing information of users and contracts
    - Nonce
    - Balance
    - CodeHash
    - StorageRoot
- Transaction
  - A unit of changing states of Klaytn blockchain platform
  - Various functions
    - Value transfer
    - Smart contract deploy
    - Smart contract execution

Account and Transaction

- Account
  - A data structure storing information of EOAs and contracts
    - Nonce
    - Balance
    - CodeHash
    - StorageRoot
- Transaction
  - A unit of changing states of Klaytn blockchain platform
  - Various functions
    - Value transfer
    - Smart contract deploy
    - Smart contract execution

For mass adoption, need better usability!

For better usability, need better acc/tx model!

Usability Considerations

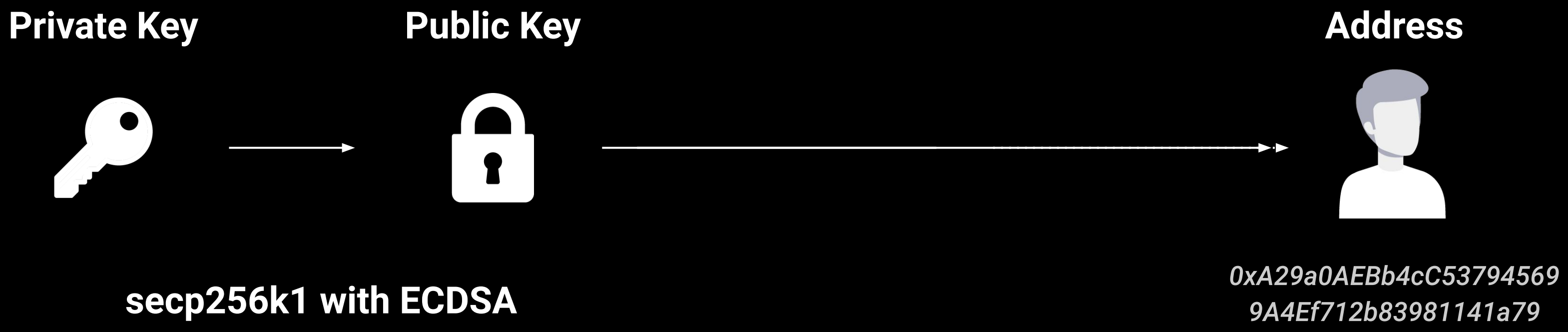
- User’s perspective
- Service provider’s perspective
- Platform developer’s perspective

Usability Considerations for Users

Usability Considerations for Users

- User’s perspective
  - **Exposed private key**
  - Increasing security of the account

Relation between Key Pair and Address



Private Key

Public Key

Address

secp256k1 with ECDSA

0xA29a0AEBb4cC53794569
9A4Ef712b83981141a79

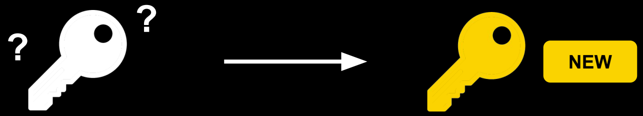
Exposed Private Key

What if your private key is exposed?

Exposed
Private Key

Exposed Private Key

What if your private key is exposed?

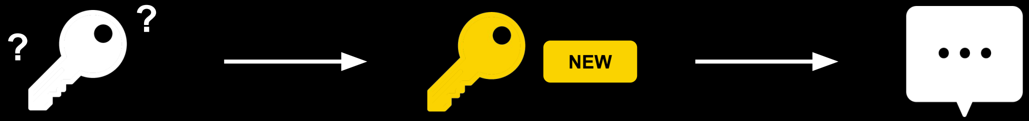


Exposed
Private Key

you need to create
another account

Exposed Private Key

What if your private key is exposed?



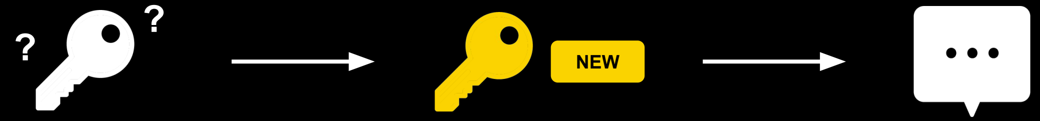
Exposed
Private Key

you need to create
another account

Tell
the new address
to your friends.

Exposed Private Key

What if your private key is exposed?



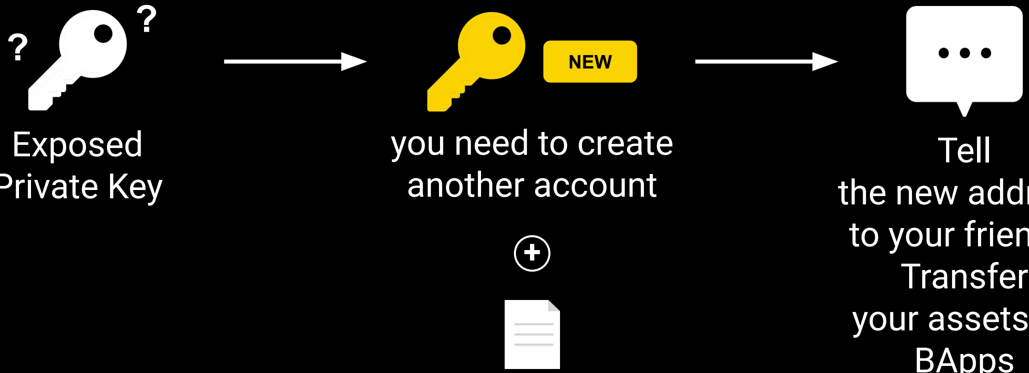
Exposed
Private Key

you need to create
another account

Tell
the new address
to your friends.
Transfer
your assets on
BApps

Exposed Private Key

What if your private key is exposed?



Exposed Private Key

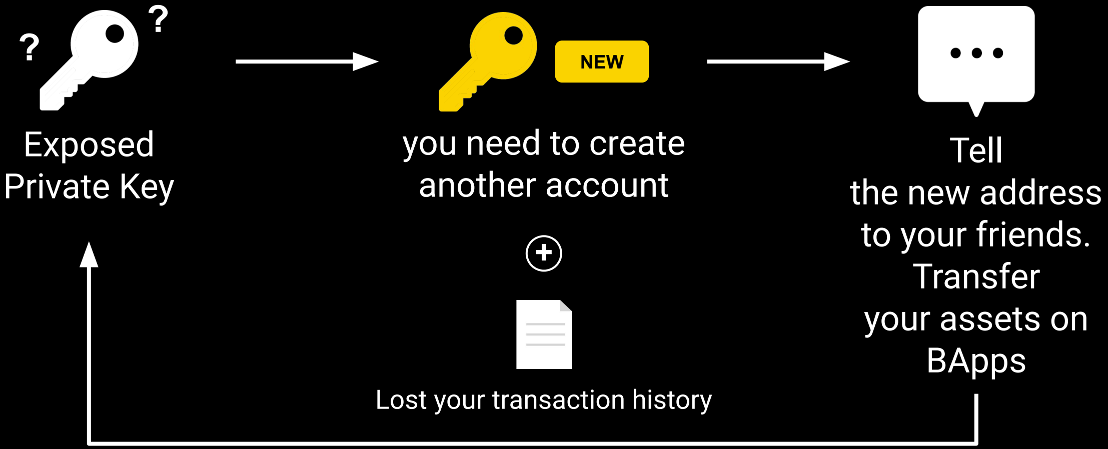
you need to create another account

Lost your transaction history

Tell the new address to your friends. Transfer your assets on BApps

Exposed Private Key

What if your private key is exposed?



Exposed Private Key

you need to create another account

NEW

Lost your transaction history

Tell the new address to your friends. Transfer your assets on BApps

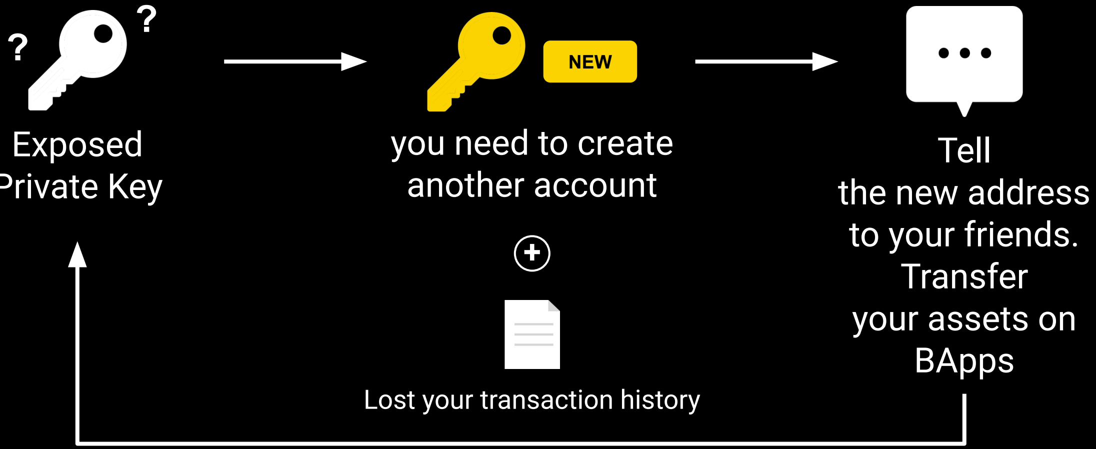
GROUND X

Exposed Private Key

Address == Bank account number

Private key == Password

What if your private key is exposed?



Exposed Private Key

you need to create another account

Lost your transaction history

Tell the new address to your friends. Transfer your assets on BApps

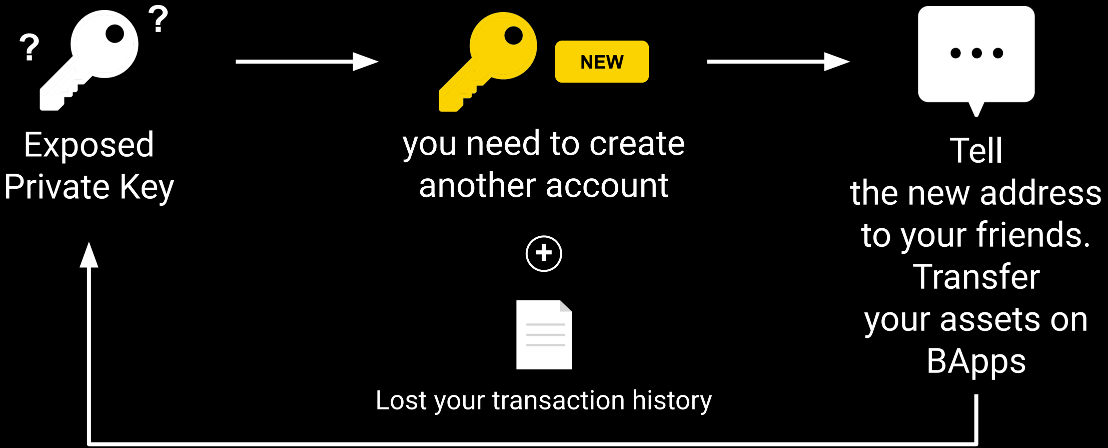
Exposed Private Key

Address == Bank account number

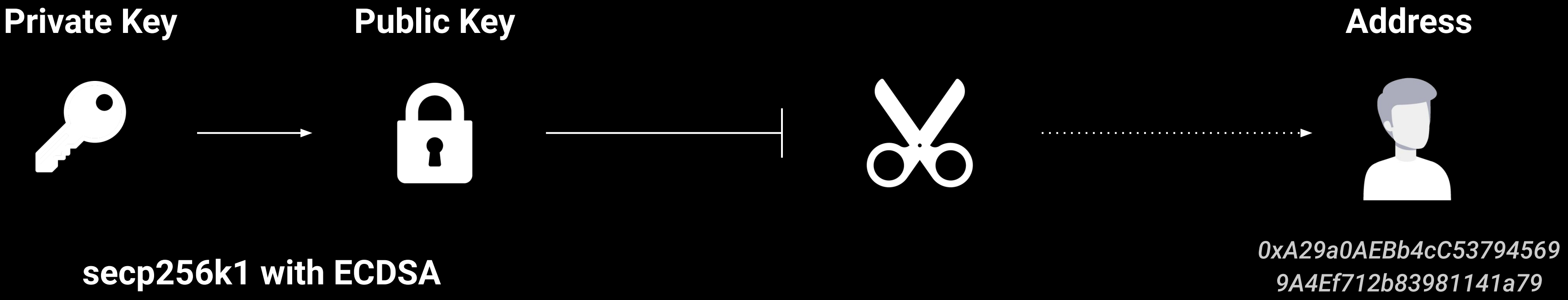
Private key == Password

What if your private key is exposed?

Solution:
Make private key changeable


? ?
Exposed Private Key
NEW
you need to create another account
Lost your transaction history
Tell the new address to your friends. Transfer your assets on BApps

Decoupling Key Pair from Address



Private Key

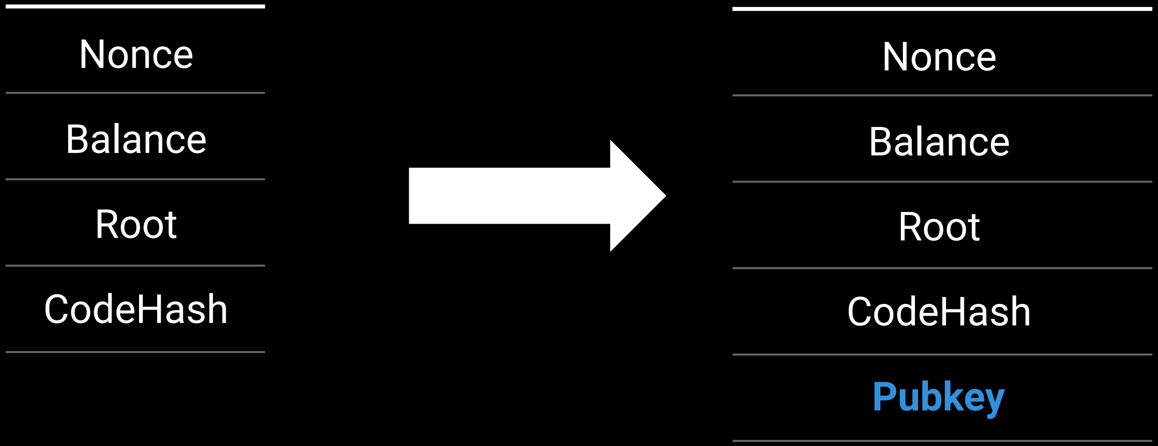
Public Key

Address

secp256k1 with ECDSA

0xA29a0AEBb4cC53794569
9A4Ef712b83981141a79

Public key in Account



Nonce

Balance

Root

CodeHash

Nonce

Balance

Root

CodeHash

Pubkey

Usability Considerations for Users

- User’s perspective
  - Exposed private key
  - **Increasing security of the account**

Increasing the Security of Your Account

- Traditional solution
  - Multisig smart contract

Increasing the Security of Your Account

- Traditional solution
  - Multisig smart contract

- Problems of multisig smart contracts
  - What is smart contract?
  - How to deploy it?
  - How to execute it?
  - How to guarantee the contract is secured?

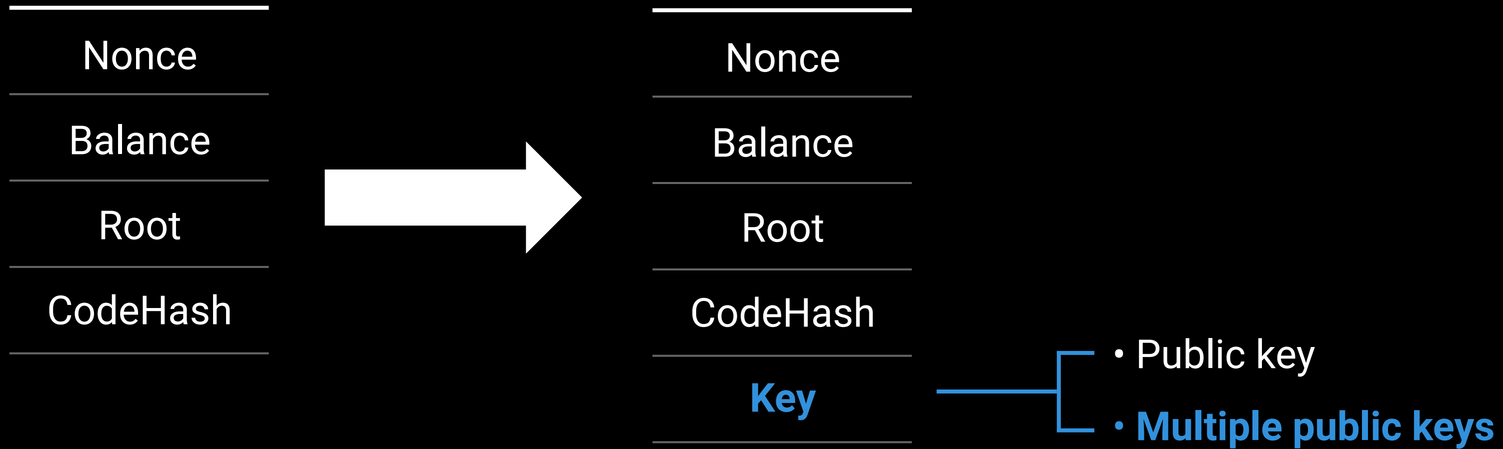
Increasing the Security of Your Account

- Traditional solution
  - Multisig smart contract

- Problems of multisig smart contracts
  - What is smart contract?
  - How to deploy it?
  - How to execute it?
  - How to guarantee the contract is secured?

- With Klaytn
  - Native support of multisig

Multisig in Account



Nonce

Balance

Root

CodeHash

Nonce

Balance

Root

CodeHash

Key

- Public key
- **Multiple public keys**

SUMMARY: Usability Considerations for Users

- User’s perspective
  - Exposed private key
  - Increasing security of the account

- Solution
  - Changeable private keys
  - Native support of multisig

Usability Considerations for Service Providers

Usability Considerations for Service Providers

- Service provider’s perspective
  - **Transaction fee**
  - Separation of permission

Fee Delegation

- Transaction fee
  - Paid on every action of a user
- Normal services
  - No fee for common actions
  - Trial period

Fee Delegation

- Transaction fee
  - Paid on every action of a user
- Normal services
  - No fee for common actions
  - Trial period
- With Klaytn
  - Transaction fee can be paid by service providers
  - Services can take various user acquisition strategies

FREE TRIAL

Fee Delegated Transactions

AccountNonce

Price

GasLimit

Recipient

Amount

Payload

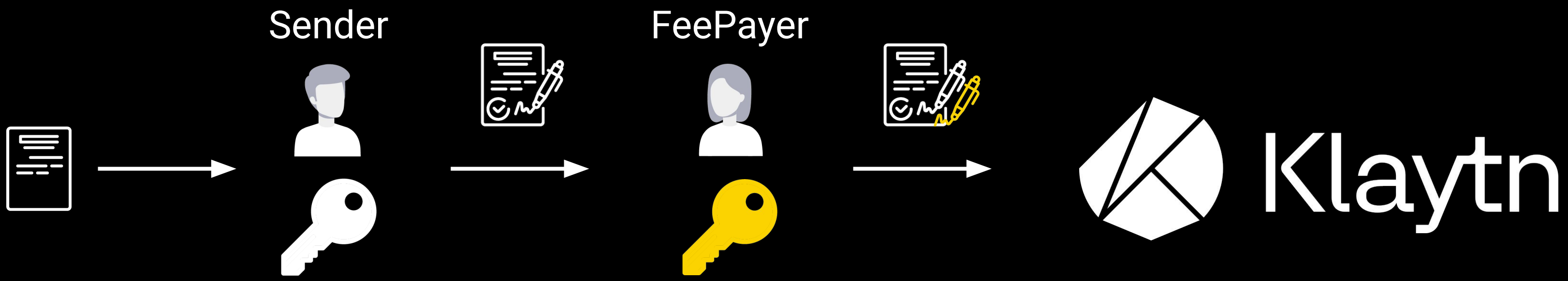
Sender address

Sender signatures

Fee payer address

Fee payer signatures

Fee Delegated Transaction Execution



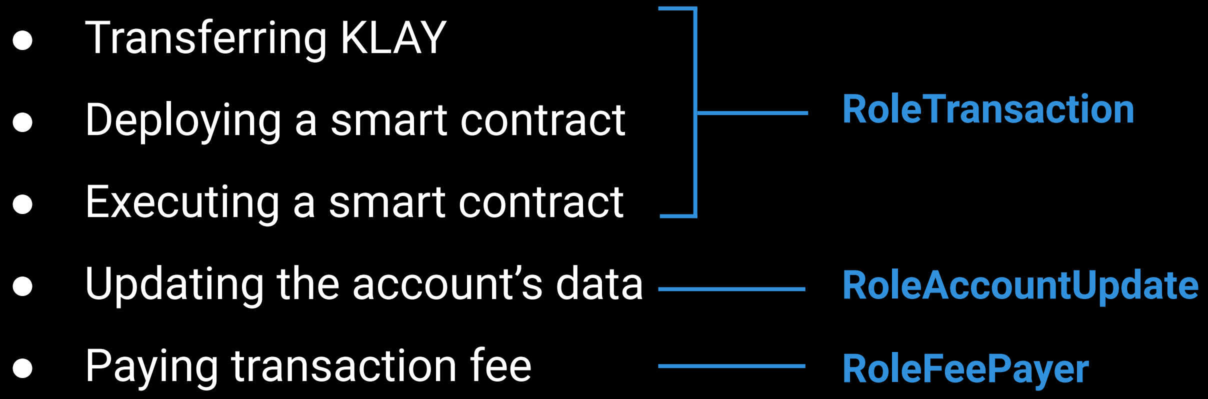
Usability Considerations for Service Providers

- Service provider’s perspective
  - Transaction fee
  - **Separation of permission**

Permissions

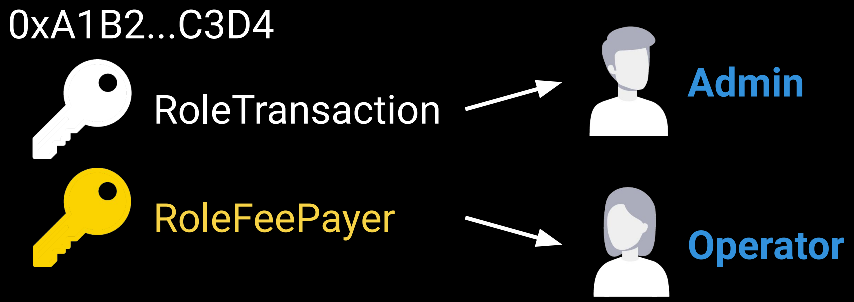
- Transferring KLAY
- Deploying a smart contract
- Executing a smart contract
- Updating the account’s data
- Paying transaction fee

Permissions and Roles



- Transferring KLAY
- Deploying a smart contract
- Executing a smart contract
- Updating the account's data
- Paying transaction fee

Role-based Key Use Case - Fee Delegation



0xA1B2...C3D4

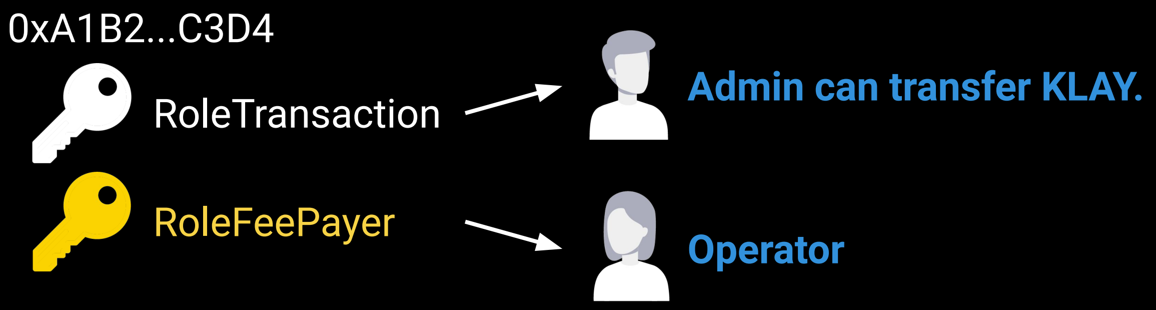
RoleTransaction

RoleFeePayer

Admin

Operator

Role-based Key Use Case - Fee Delegation

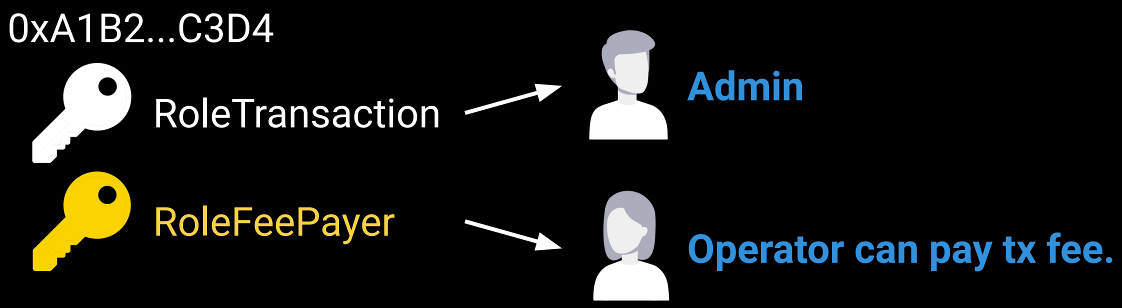


0xA1B2...C3D4

RoleTransaction → Admin can transfer KLAY.

RoleFeePayer → Operator

Role-based Key Use Case - Fee Delegation

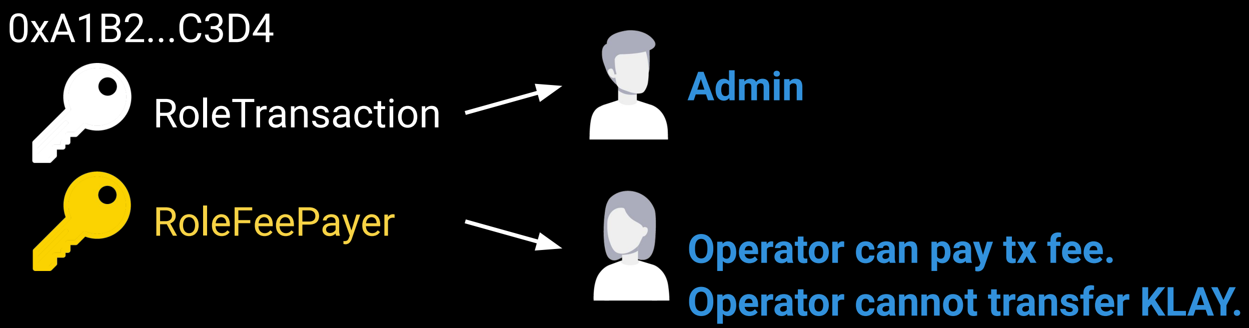


0xA1B2...C3D4

RoleTransaction → Admin

RoleFeePayer → Operator can pay tx fee.

Role-based Key Use Case - Fee Delegation

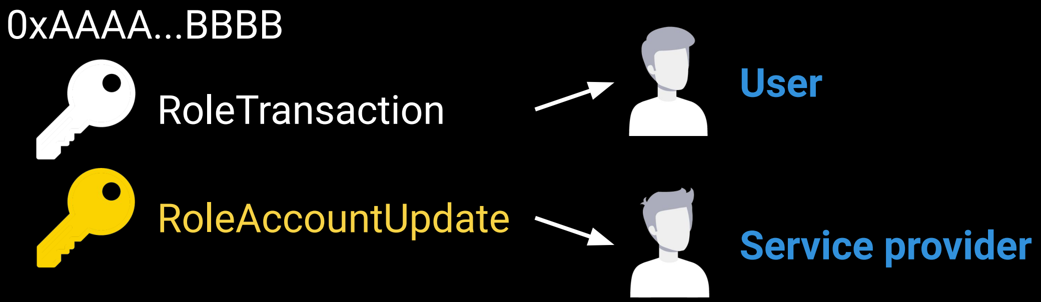


0xA1B2...C3D4

RoleTransaction → Admin

RoleFeePayer → Operator can pay tx fee.
Operator cannot transfer KLAY.

Role-based Key Use Case - User Account Recovery

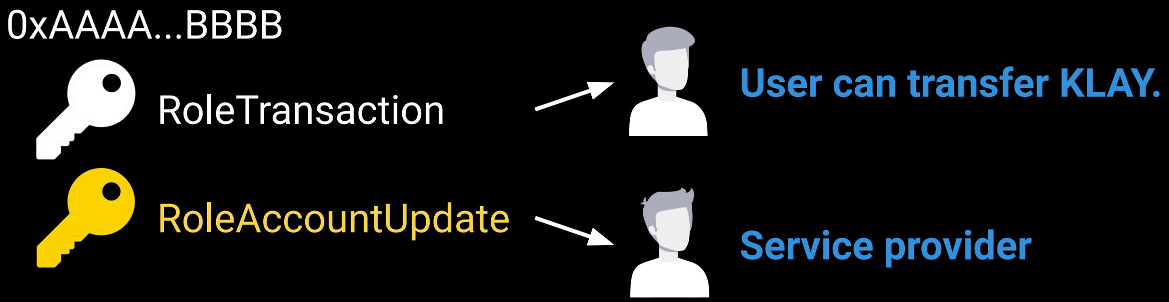


0xAAAA...BBBB

RoleTransaction → User

RoleAccountUpdate → Service provider

Role-based Key Use Case - User Account Recovery

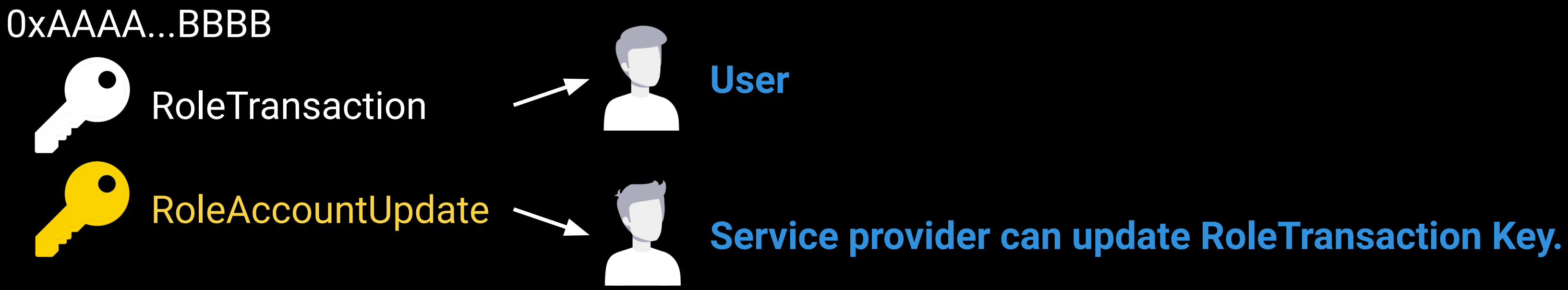


0xAAAA...BBBB

RoleTransaction → User can transfer KLAY.

RoleAccountUpdate → Service provider

Role-based Key Use Case - User Account Recovery



0xAAAA...BBBB

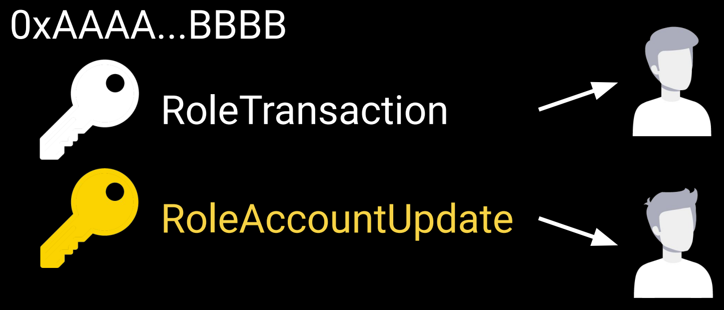
RoleTransaction

RoleAccountUpdate

User

Service provider can update RoleTransaction Key.

Role-based Key Use Case - User Account Recovery



0xAAAA...BBBB

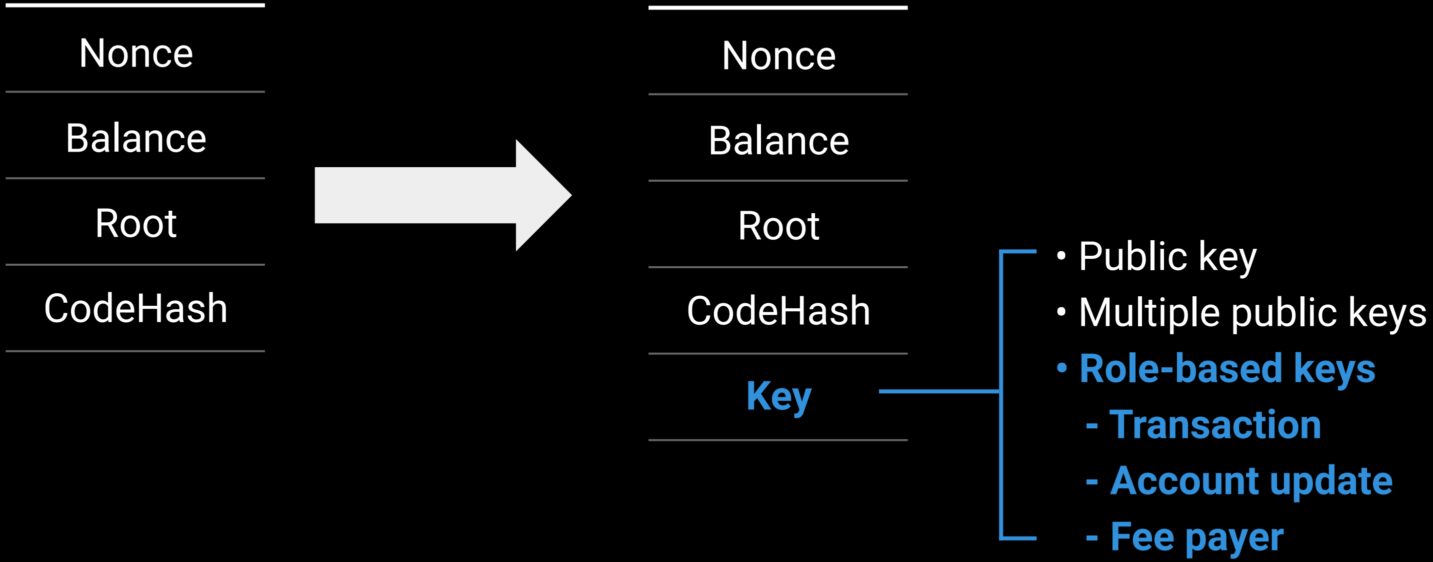
RoleTransaction

RoleAccountUpdate

User

Service provider can update RoleTransaction Key.
Service provider can give a new key to the user.

Role-based Key in Account



Nonce

Balance

Root

CodeHash

Nonce

Balance

Root

CodeHash

Key

- Public key
- Multiple public keys
- **Role-based keys**
  - **Transaction**
  - **Account update**
  - **Fee payer**

SUMMARY: Usability Considerations for Service Providers

- Service provider’s perspective
  - Transaction fee
  - Separation of permission

- Solution
  - Fee Delegated transactions
  - Native support of role-based keys

Usability Considerations for Platform Developers

GROUND X

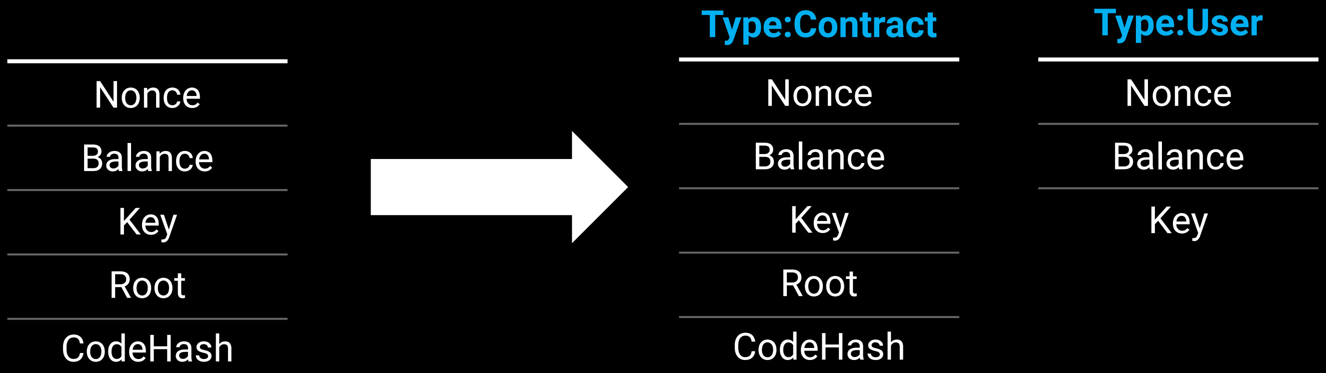
Usability Considerations For Platform Developers

- Platform developer’s perspective
  - Easy to extend
  - Easy to analyze

Account Type

Nonce
Balance
Key
Root
CodeHash

Account Type



Nonce

Balance

Key

Root

CodeHash

Type:Contract

Nonce

Balance

Key

Root

CodeHash

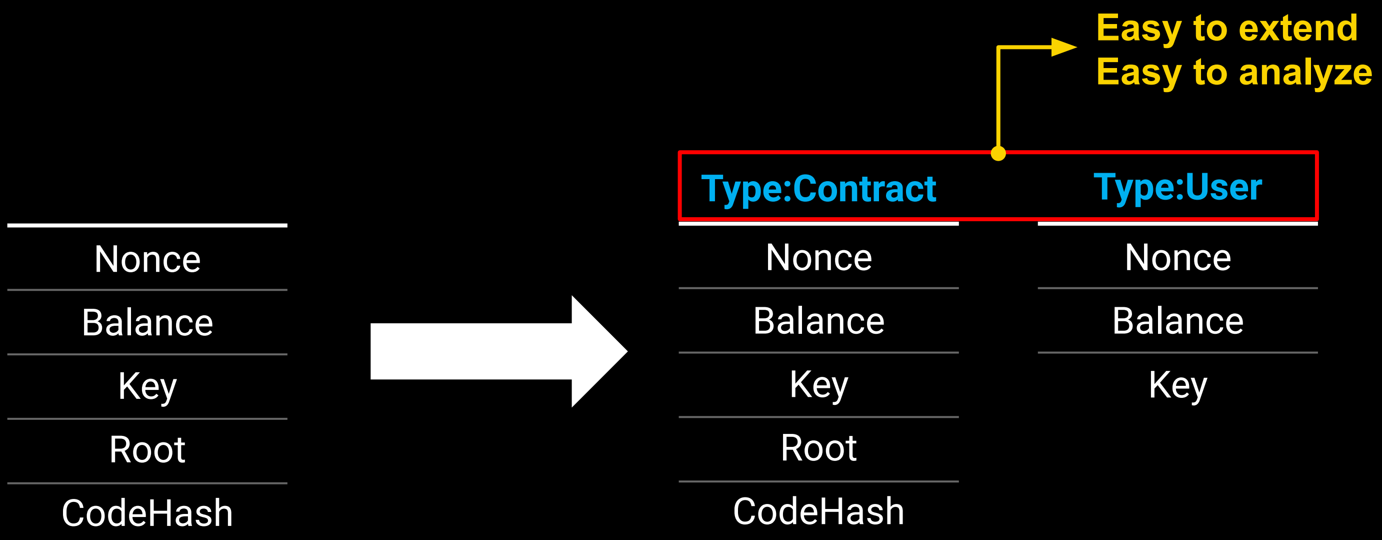
Type:User

Nonce

Balance

Key

Account Type



Nonce

Balance

Key

Root

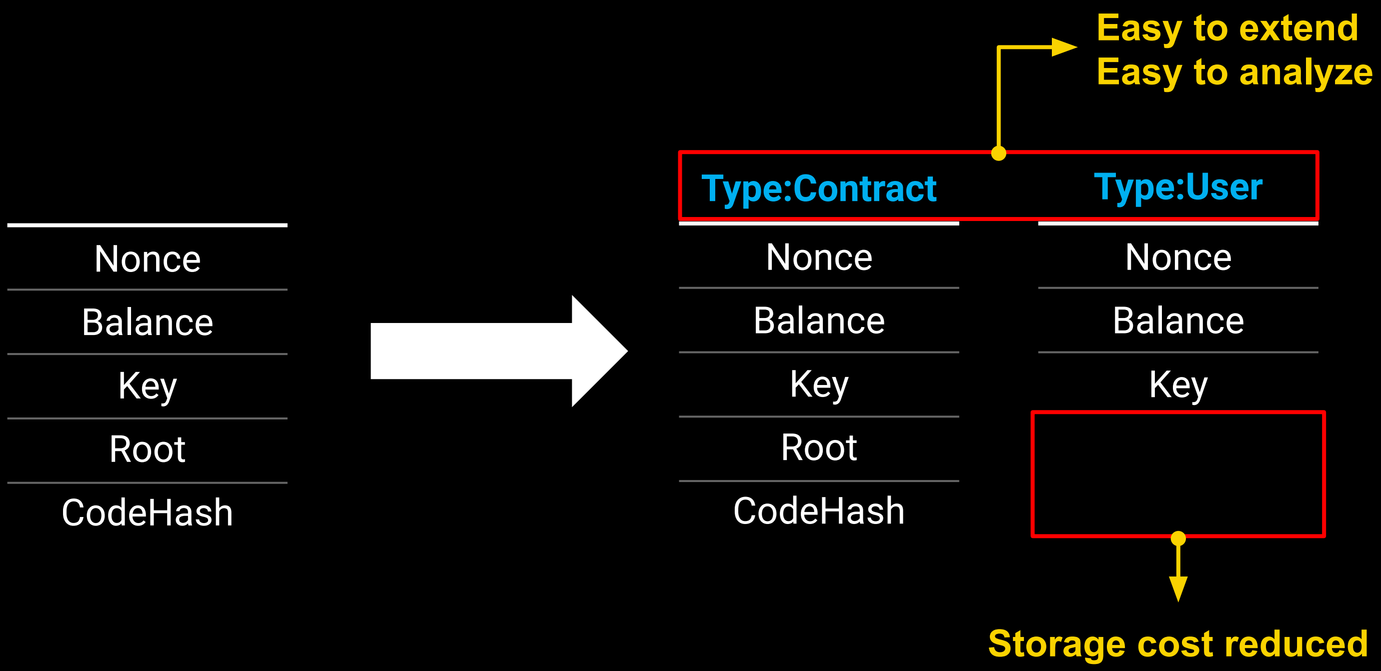
CodeHash

Easy to extend

Easy to analyze

Type:Contract	Type:User
Nonce	Nonce
Balance	Balance
Key	Key
Root	
CodeHash	

Account Type



Nonce

Balance

Key

Root

CodeHash

Easy to extend
Easy to analyze

Type:Contract

Nonce

Balance

Key

Root

CodeHash

Type:User

Nonce

Balance

Key

Storage cost reduced

Transaction Type

AccountNonce

Price

GasLimit

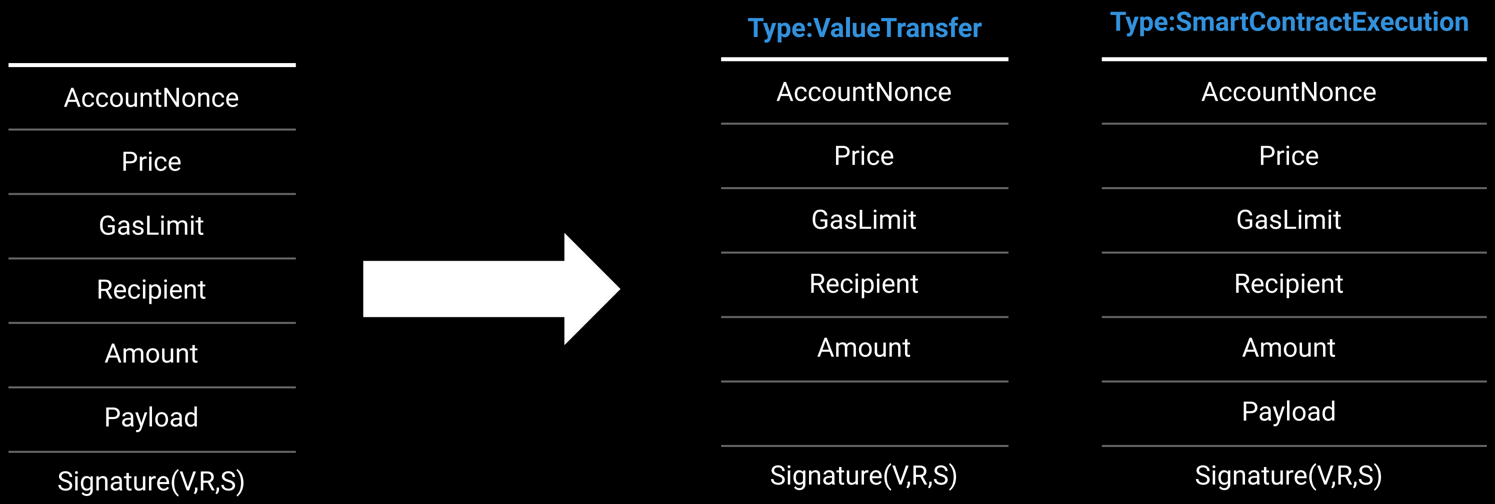
Recipient

Amount

Payload

Signature(V,R,S)

Transaction Type



AccountNonce

Price

GasLimit

Recipient

Amount

Payload

Signature(V,R,S)

Type:ValueTransfer

AccountNonce

Price

GasLimit

Recipient

Amount

Signature(V,R,S)

Type:SmartContractExecution

AccountNonce

Price

GasLimit

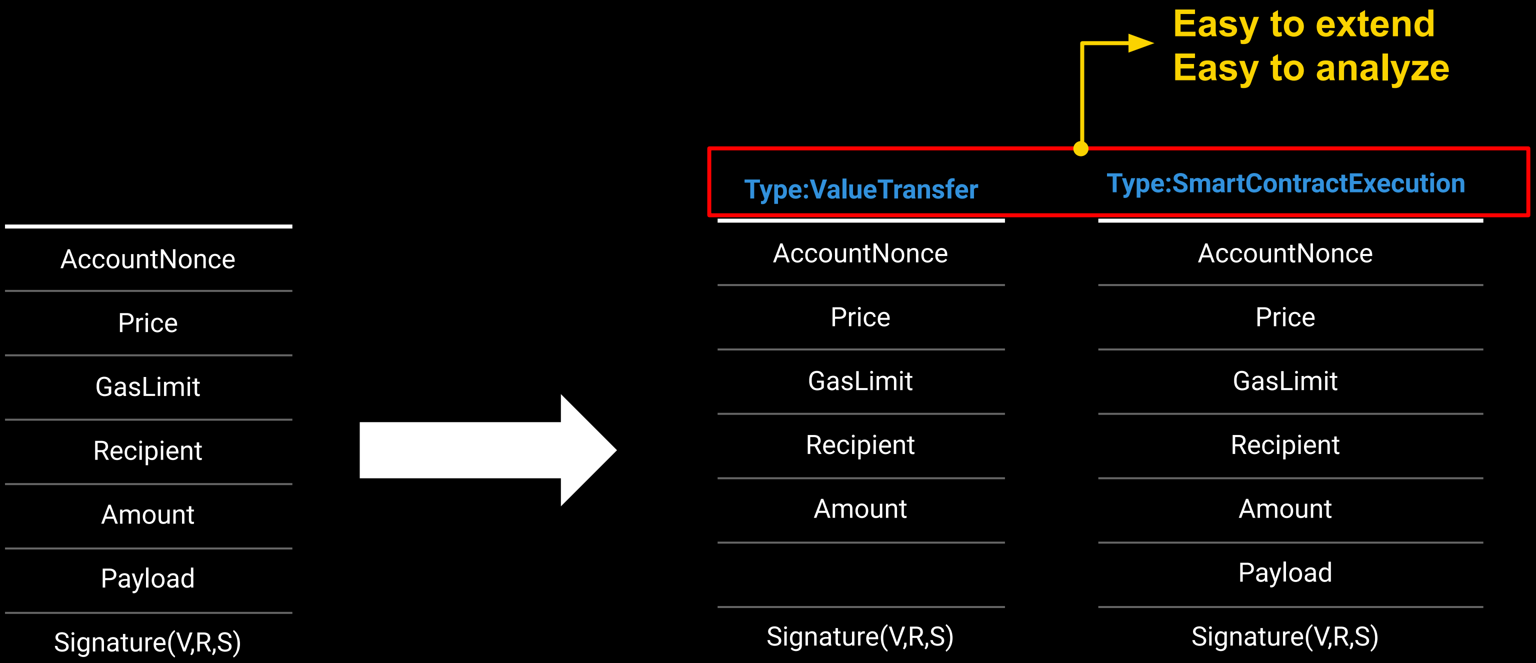
Recipient

Amount

Payload

Signature(V,R,S)

Transaction Type

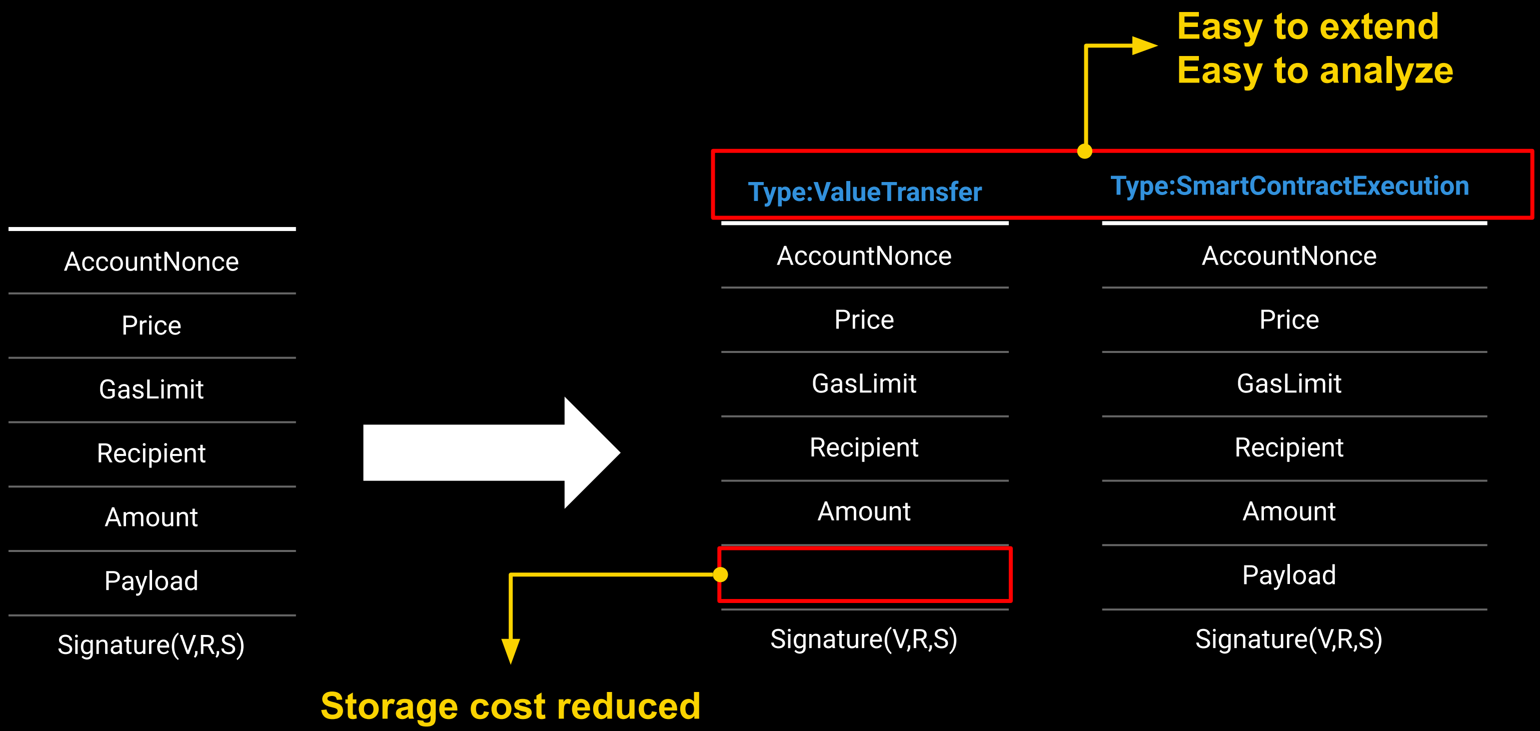


AccountNonce
Price
GasLimit
Recipient
Amount
Payload
Signature(V,R,S)

Type:ValueTransfer	Type:SmartContractExecution
AccountNonce	AccountNonce
Price	Price
GasLimit	GasLimit
Recipient	Recipient
Amount	Amount
	Payload
Signature(V,R,S)	Signature(V,R,S)

Easy to extend
Easy to analyze

Transaction Type



AccountNonce

Price

GasLimit

Recipient

Amount

Payload

Signature(V,R,S)

Type:ValueTransfer

AccountNonce

Price

GasLimit

Recipient

Amount

Signature(V,R,S)

Type:SmartContractExecution

AccountNonce

Price

GasLimit

Recipient

Amount

Payload

Signature(V,R,S)

Easy to extend
Easy to analyze

Storage cost reduced

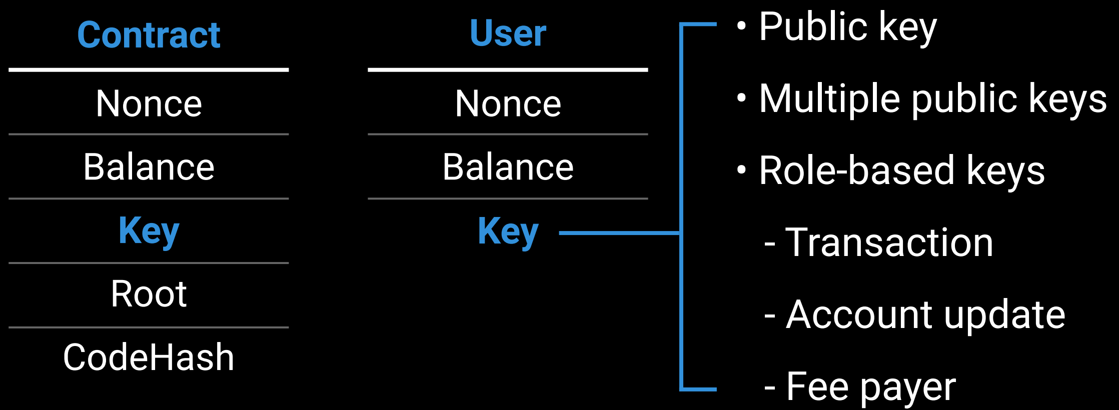
SUMMARY: Usability Considerations For Platform Developers

- Platform developer’s perspective
  - Easy to extend
  - Easy to analyze

- Solution
  - Introduce explicit type fields to
    - accounts
    - transactions

Account Model

Account Model



Contract	User
Nonce	Nonce
Balance	Balance
Key	**Key**
Root	
CodeHash	

- Public key
- Multiple public keys
- Role-based keys
  - Transaction
  - Account update
  - Fee payer

Transaction Model

Transaction Model

	Basic	Fee Delegation
Legacy	TxTypeLegacyTransaction	N/A
ValueTransfer	TxTypeValueTransfer	TxTypeFeeDelegatedValueTransfer
ValueTransferMemo	TxTypeValueTransferMemo	TxTypeFeeDelegatedValueTransferMemo
SmartContractDeploy	TxTypeSmartContractDeploy	TxTypeFeeDelegatedSmartContractDeploy
SmartContractExecution	TxTypeSmartContractExecution	TxTypeFeeDelegatedSmartContractExecution
AccountUpdate	TxTypeAccountUpdate	TxTypeFeeDelegatedAccountUpdate
...	...	...

https://docs.klaytn.com/klaytn/design/transactions

Transaction Model

Functionality ↓

	Basic	Fee Delegation
Legacy	TxTypeLegacyTransaction	N/A
ValueTransfer	TxTypeValueTransfer	TxTypeFeeDelegatedValueTransfer
ValueTransferMemo	TxTypeValueTransferMemo	TxTypeFeeDelegatedValueTransferMemo
SmartContractDeploy	TxTypeSmartContractDeploy	TxTypeFeeDelegatedSmartContractDeploy
SmartContractExecution	TxTypeSmartContractExecution	TxTypeFeeDelegatedSmartContractExecution
AccountUpdate	TxTypeAccountUpdate	TxTypeFeeDelegatedAccountUpdate
…	…	…

https://docs.klaytn.com/klaytn/design/transactions

Transaction Model

Fee delegation →

Functionality ↓

	Basic	Fee Delegation
Legacy	TxTypeLegacyTransaction	N/A
ValueTransfer	TxTypeValueTransfer	TxTypeFeeDelegatedValueTransfer
ValueTransferMemo	TxTypeValueTransferMemo	TxTypeFeeDelegatedValueTransferMemo
SmartContractDeploy	TxTypeSmartContractDeploy	TxTypeFeeDelegatedSmartContractDeploy
SmartContractExecution	TxTypeSmartContractExecution	TxTypeFeeDelegatedSmartContractExecution
AccountUpdate	TxTypeAccountUpdate	TxTypeFeeDelegatedAccountUpdate
…	…	…

https://docs.klaytn.com/klaytn/design/transactions

What's Next?

- Human-readable address
  - 0x1234...CDEF -> colin.klaytn
- More account types
- More transaction types

Conclusion

Conclusion

- Design account and transaction model to enhance usability

- Users
  - Changeable private key
  - Native support of multisig

- Service providers
  - Fee delegation
  - Native support of role-based keys

- Platform Developers
  - Explicit types for accounts and transactions

Find more: https://medium.com/@klaytn.tech

Something More!

Contribute!

- Klaytn organization in Github : https://github.com/klaytn

Klaytn	https://github.com/klaytn/klaytn
caver-js	https://github.com/klaytn/caver-js
caver-java	https://github.com/klaytn/caver-java
Klaytn Improvement Proposal (KIP)	https://github.com/klaytn/kips

WE ARE HIRING!

https://www.groundx.xyz/careers

THANK YOU

Ground X
27F, 521, Teheran-ro,
Gangnam-gu, Seoul, Republic of Korea