東莞理工學院
DONGGUAN UNIVERSITY OF TECHNOLOGY

区块链技术与应用

v0.10.26

实验一：分布式账本

丁烨，网络空间安全学院 副教授
dingye@dgut.edu.cn

分布式账本概述

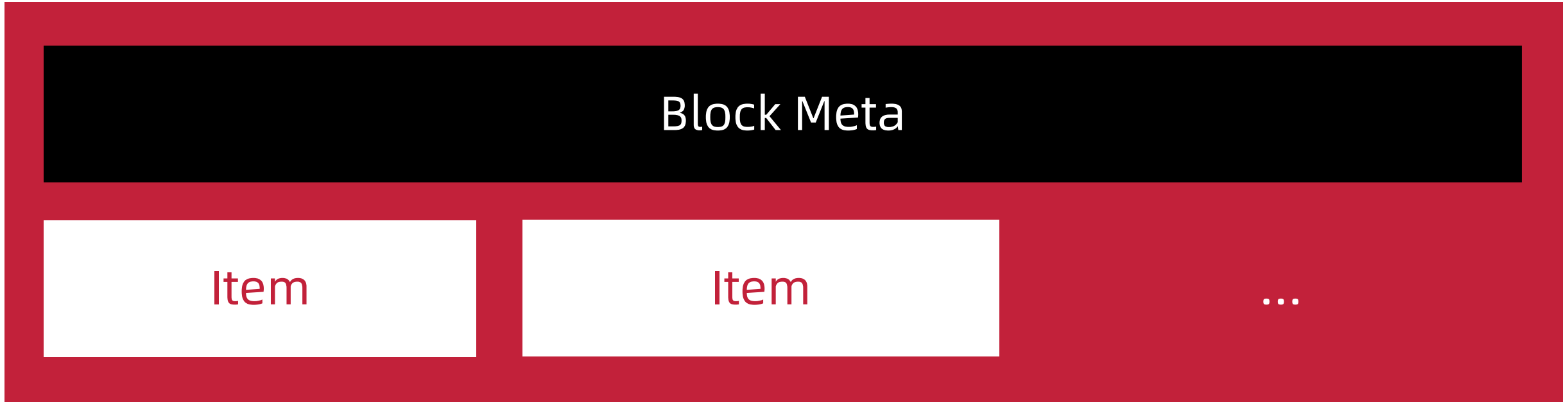
区块链核心思想

- 区块链（Blockchain）本质上是一个数据库
- 该数据库的设计目的为：安全、防止篡改
- 但是，区块链为了安全付出了运行缓慢的代价

分布式账本概述

区块链核心思想

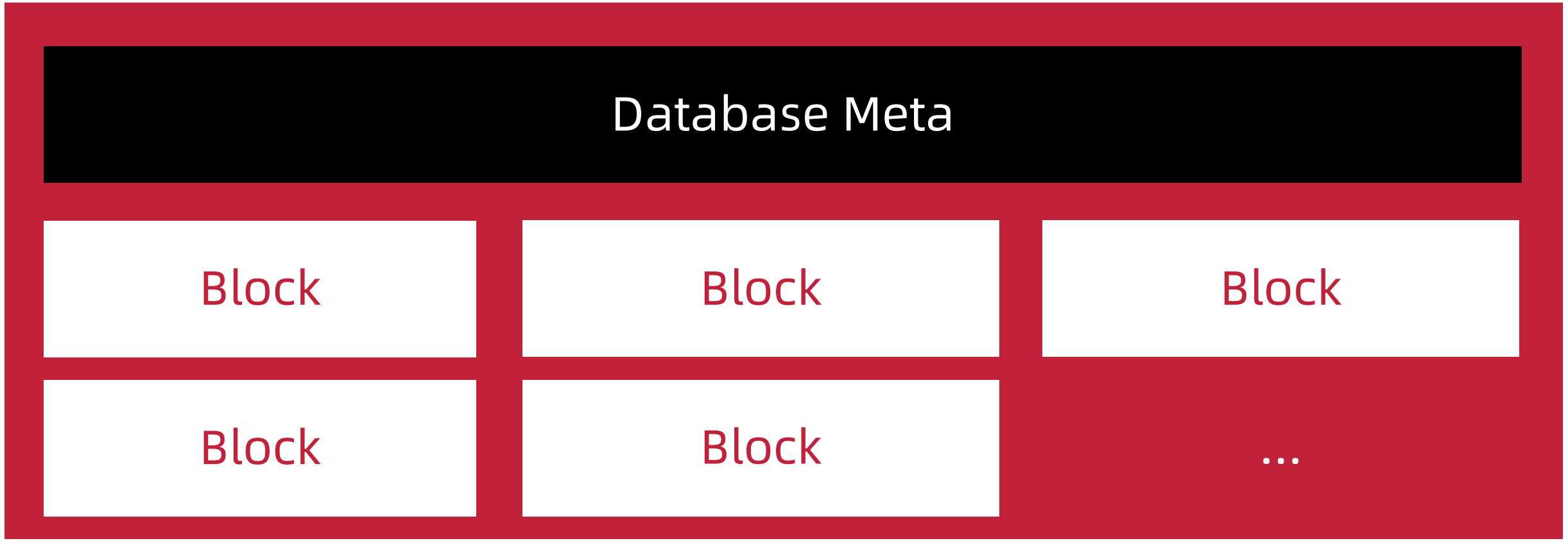
- “区块”（Block）



分布式账本概述

区块链核心思想

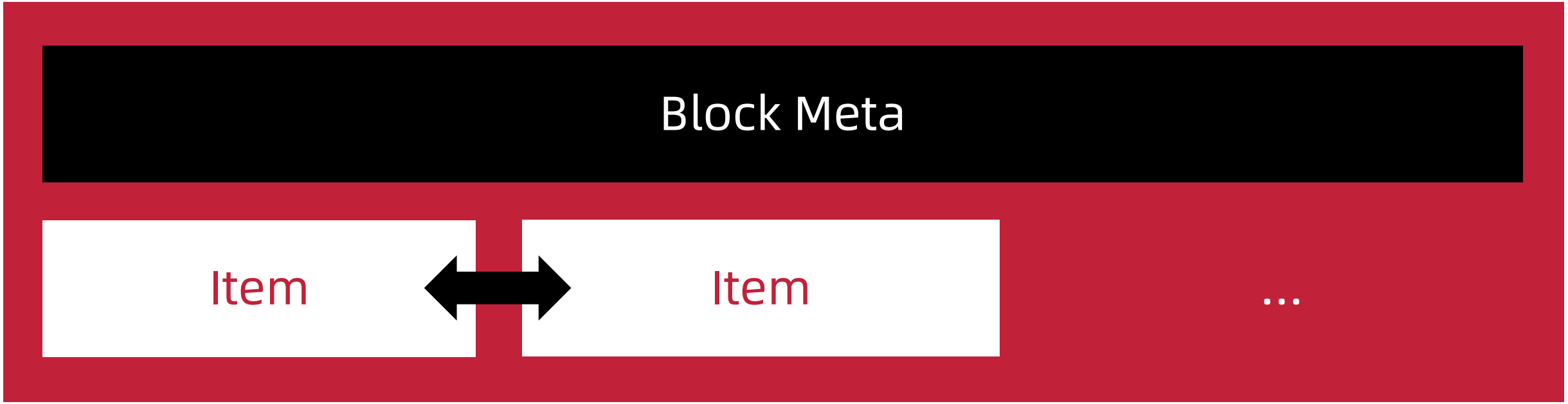
- “区块”（Block）



分布式账本概述

区块链核心思想

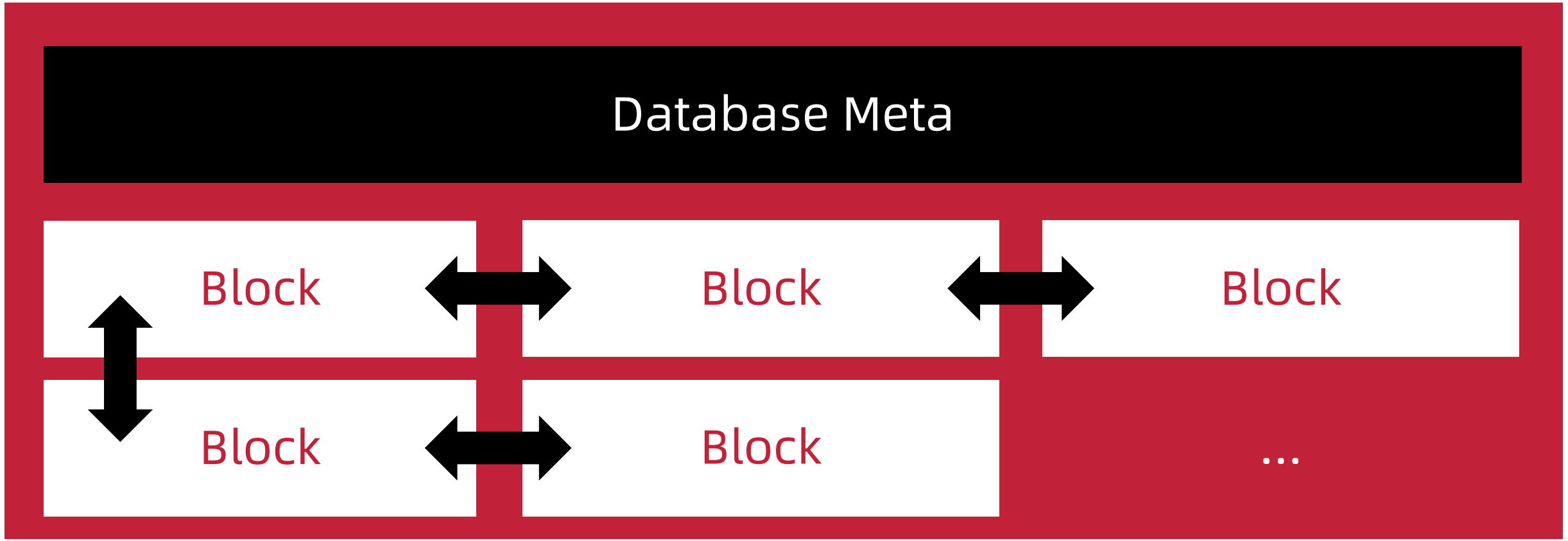
- “链”（Chain）



分布式账本概述

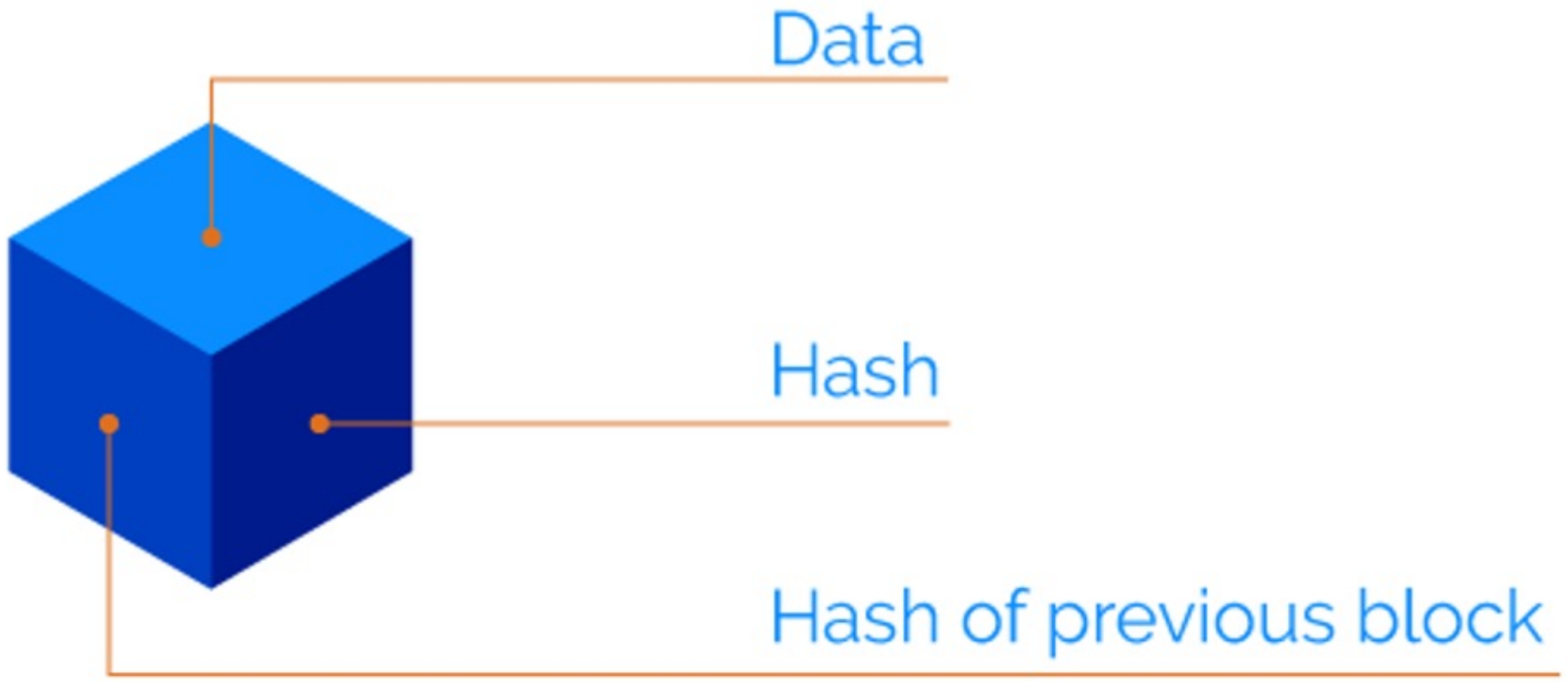
区块链核心思想

- “链”（Chain）



分布式账本概述

区块链核心思想



分布式账本概述

金融账本基础

- 交易（Transaction）
- 又称贸易、交换、互市，是买卖双方对有价物品及服务进行互通有无的行为
- 可以是以货币为交易媒介的过程，也可以是以物易物
- 在区块链账本系统中，一条记录（Item）通常为一条交易（Transaction）

分布式账本概述

金融账本基础

- 账本（Ledger）
- 具有一定格式与若干账页组成，以会计凭证为依据，对所有经济业务进行序时分类记录的本籍，也就是通常我们所说的账册
- 简单来讲，一个账本包含多条交易记录
- 在区块链账本系统中，一个账本（Ledger）即为一个区块链（Blockchain）
- 区块（Block）相当于为账本的分页（Page）

分布式账本概述

分布式账本特点

- 分布式账本（Distributed Ledger）
- 又称共享账本（Shared Ledger）、分散式账本技术（Distributed ledger Technology，DLT）
- 是一个由多站点、多国家或多家机构所组成的在网络上进行电子数据复制、共享及同步的共识机制
- 通常不依赖也不存在中央管理员或集中的数据存储
- 对等网络与共识机制确保了跨节点间的数据能被正确复制
- 区块链系统是分布式账本的一种实现方法

实现简单的分布式账本

编程基础

https://websitesetup.org/wp-content/uploads/2020/04/Python-Cheat-Sheet.pdf
https://perso.limsi.fr/pointal/_media/python:cours:mementopython3-english.pdf

实现简单的分布式账本

可做散列函数的基类

❖ basic.py

```
import hashlib
from wrenchbox.object import Dict2StrSafe

class Base(Dict2StrSafe):
    def hash(self):
        return hashlib.sha1(str(self.__dict__).encode()).hexdigest()
```

实现简单的分布式账本

交易记录类

❖ transaction.py

```
from datetime import datetime
from uuid import UUID

from .basic import Base

class Transaction(Base):
    def __init__(self,
                 sender: str, receiver: str,
                 amount: int, t: float = None,
                 prev_hash: str = None):
        ...
```

实现简单的分布式账本

交易记录类

❖ transaction.py

```
class Transaction(Base):
    def __init__(self, …):
        assert UUID(sender, version=4)
        assert UUID(receiver, version=4)
        self.sender = sender
        self.receiver = receiver
        self.amount = amount
        self.t = t if t is not None else datetime.now().timestamp()
        self.prev_hash = prev_hash
```

实现简单的分布式账本

区块链类

❖ blockchain.py

```
from .basic import Base

class Block(Base):
    def __init__(self, prev_hash: str = None):
        self.items = []
        self.prev_hash = prev_hash

    ...
```

实现简单的分布式账本

区块链类

- blockchain.py

```
class Block(Base):
    ...

    def add(self, item):
        item.prev_hash = self.items[-1].hash() if len(self.items) else None
        self.items.append(item)

    ...
```

实现简单的分布式账本

区块链类

- blockchain.py

```
class Block(Base):
    ...

    def validate(self):
        for i in range(len(self.items)):
            assert i == 0 or self.items[i].prev_hash == self.items[i - 1].hash()
```

实现简单的分布式账本

区块链类

- blockchain.py

```
class Blockchain(Base):
    def __init__(self):
        self.blocks = []

    def add(self, block):
        block.prev_hash = self.blocks[-1].hash() if len(self.blocks) else None
        self.blocks.append(block)

    ...
```

实现简单的分布式账本

区块链类

- blockchain.py

```
class Blockchain(Base):
    ...

    def validate(self):
        for i in range(len(self.blocks)):
            assert i == 0 or self.blocks[i].prev_hash == self.blocks[i - 1].hash()
            self.blocks[i].validate()
```

实现简单的分布式账本

测试本地账本

❖ test.py

```
import json
import random
from uuid import uuid4

from foxchain.blockchain import Blockchain, Block
from foxchain.transaction import Transaction

if __name__ == '__main__':
    chain = Blockchain()
    ...
```

实现简单的分布式账本

测试本地账本

- test.py

```
...
if __name__ == '__main__':
    chain = Blockchain()
    for i in range(3):
        block = Block()
        for j in range(10):
            transaction = Transaction(sender=str(uuid4()), receiver=str(uuid4()),
                amount=random.randint(1, 1000000)
            )
            block.add(transaction)
        chain.add(block)
    ...
```

实现简单的分布式账本

测试本地账本

❖ test.py

```
...
if __name__ == '__main__':
    ...
    print(json.dumps(json.loads(str(chain)), indent=4, sort_keys=True))
    print(chain.validate())
    ...
```

```
~/Downloads/foxchain-master » python3 test.py                    valency@Aorus-Master
{
    "blocks": [
        {
            "items": [
                {
                    "amount": 356347,
                    "prev_hash": null,
                    "receiver": "37ef4c46-ff77-403f-9e6d-98ba2bf1dee6",
                    "sender": "fcad5575-b26c-428b-9838-19ffd61550d2",
                    "t": 1616080992.596395
                },
                {
                    "amount": 81010,
                    "prev_hash": "a591128e038be3e10ee2bcf9a5958d4161161aca",
                    "receiver": "0aec5738-92d5-4476-95c7-adadbd2a2fc3",
                    "sender": "c477faa2-745c-44af-bc13-6473161022b6",
                    "t": 1616080992.596419
                },
```

```
            "amount": 176919,
            "prev_hash": "ca2ebeb7800be3c9ed5ceb71cc9d2a2b4afc051e",
            "receiver": "68e197cf-6d78-408d-8930-16be4de080f6",
            "sender": "e2ad571f-bf7c-4aef-a440-e4cfe4c93359",
            "t": 1616080992.596561
        }
    ],
    "prev_hash": null
},
{
    "items": [
        {
            "amount": 561962,
            "prev_hash": null,
            "receiver": "2b2f739e-ac64-4989-9e93-51441319436a",
            "sender": "e076bf94-c427-4b69-b8a8-e05f72b1efad",
            "t": 1616080992.596578
        },
        {
            "amount": 62726,
```

```
            {
                "amount": 562685,
                "prev_hash": "569db9a871dbb5a05f39c23a105f1749ef91286b",
                "receiver": "f0ba9efa-6d18-4070-a213-f76de5056a96",
                "sender": "602fb937-14e0-4706-90f9-348511f4fe53",
                "t": 1616080992.596876
            },
            {
                "amount": 661301,
                "prev_hash": "e86c661515d935a0234c7c229b48e20b37ae3a66",
                "receiver": "fb564195-9bdd-4367-90d3-b22da9392174",
                "sender": "c62d172c-dbc1-4fb3-b015-c90786eac802",
                "t": 1616080992.596894
            }
        ],
        "prev_hash": "f830a6e33f652408d07a2ade01a922f36f6f50e8"
    }
]
}
None
```


实现简单的分布式账本

测试本地账本

❖ test.py

```
...
if __name__ == '__main__’:
    ...
    chain.blocks[0].items[0].receiver = str(uuid4())
    print(json.dumps(json.loads(str(chain)), indent=4, sort_keys=True))
    print(chain.validate())
```

```
                    "receiver": "aa77a86d-0f26-45c4-83ed-64426782cad5",
                    "sender": "7a342e40-4a54-41a0-8b07-347cc3b8bcc9",
                    "t": 1616081241.470215
                }
            ],
            "prev_hash": "964b2eadebde2ad33a52c4219b79fd2b6e198559"
        }
    ]
}
```


```
Traceback (most recent call last):
  File "test.py", line 24, in <module>
    print(chain.validate())
  File "/mnt/c/Users/Valency/Downloads/foxchain-master/foxchain/blockchain.py", line 29, in
validate
    self.blocks[i].validate()
  File "/mnt/c/Users/Valency/Downloads/foxchain-master/foxchain/blockchain.py", line 15, in
validate
    assert i == 0 or self.items[i].prev_hash == self.items[i - 1].hash()
AssertionError
--------------------------------------------------------------------------------
```

作业

- 撰写一个 Ledger 类继承 Blockchain 类
- 为 Ledger 类提供一个函数，该函数能够计算指定账户的余额
- 重载 Ledger 类的添加区块函数，使得 Ledger 类在添加区块时，校验每一笔交易中的转账金额不能大于账户余额

- 提供完整的测试代码及截图
- 测试用例应当包含出错的情况

作业

```
~/Workspace/foxchain(master*) » python3 -u -m foxchain.tests.ledger                    valency@Sakura
{"sender": "00000000-0000-0000-0000-000000000000", "receiver": "7dd5fcb0-3570-4e04-96e1-737712485d90", "amount": 218250, "t
": 1616149936.120172, "prev_hash": null}
218250 0
{"sender": "7dd5fcb0-3570-4e04-96e1-737712485d90", "receiver": "654b43c2-7788-4a11-9afe-3e98a2d74d7c", "amount": 202551, "t
": 1616149936.120519, "prev_hash": null}
15699 202551
~/Workspace/foxchain(master*) » python3 -u -m foxchain.tests.ledger                    valency@Sakura
{"sender": "00000000-0000-0000-0000-000000000000", "receiver": "20a806bf-e396-4ad2-966e-31fd490fd80b", "amount": 450185, "t
": 1616149936.907113, "prev_hash": null}
450185 0
{"sender": "20a806bf-e396-4ad2-966e-31fd490fd80b", "receiver": "5378c9b1-c94e-4501-9e68-f01f48cb3964", "amount": 328853, "t
": 1616149936.907495, "prev_hash": null}
121332 328853
```

作业

```
~/Workspace/foxchain(master*) » python3 -u -m foxchain.tests.ledger                    valency@Sakura
{"sender": "00000000-0000-0000-0000-000000000000", "receiver": "43f81799-8244-4e05-bf61-88f4a27871b4", "amount": 198805, "t
": 1616149937.674487, "prev_hash": null}
198805 0
{"sender": "43f81799-8244-4e05-bf61-88f4a27871b4", "receiver": "40d3afbb-521c-440c-b5f4-3f15e2f58f7a", "amount": 898042, "t
": 1616149937.675232, "prev_hash": null}
Traceback (most recent call last):
  File "/usr/lib/python3.8/runpy.py", line 194, in _run_module_as_main
    return _run_code(code, main_globals, None,
  File "/usr/lib/python3.8/runpy.py", line 87, in _run_code
    exec(code, run_globals)
  File "/mnt/e/Workspace/foxchain/foxchain/tests/ledger.py", line 20, in <module>
    ledger.add(block)
  File "/mnt/e/Workspace/foxchain/foxchain/ledger.py", line 19, in add
    assert transaction.sender == GENESIS or self.balance(transaction.sender) ≥ transaction.amount
AssertionError
```

作业

- 在作业系统中下载并完成本实验课对应实验报告
- https://hw.css.dgut.edu.cn/
- 注意：所有标识为 * 的地方都需要填写
- 截止日期：2021-11-04 23:59

课程名称：区块链技术与应用　　学期：2021 年春季

实验名称	分布式账本			实验序号	1
姓　名	***	学　号	***	班　级	***
实验地点	***	实验日期	***	指导老师	丁烨
教师评语	-			实验成绩	-
				百分制	100
同组同学					

四、　实验作业及分析

4.1　实验过程

1) *** 请将详细实验过程填写在此处 ***

4.2　实验结果

*** 请将实验结果截图填写在此处 ***

五、　实验总结

*** 请撰写一段 200 字左右的实验总结 ***


GOOD
LUCK!