



北京大学

PEKING UNIVERSITY

区块链课程

孙惠平

sunhp@ss.pku.edu.cn



北京大学 软件与微电子学院

School of Software and Microelectronics, Peking University



北京大学
PEKING UNIVERSITY

PART 第五章

蚂蚁链技术及应用

目录

CONTENTS



- 01. 蚂蚁链简介
- 02. 蚂蚁链技术
-
- 03. 蚂蚁链产业实践
-
- 04. 蚂蚁链开发者实验室



北京大學
PEKING UNIVERSITY

PART —

蚂蚁链简介

蚂蚁科技：聚焦于面向未来的科技创新BASIC

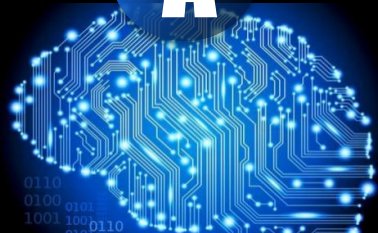
B



Blockchain 区块链

让金融服务透明可信
可审计可追溯
无需人工介入

A



AI 人工智能

极大提升数据处理效率
提升智能决策水平
拓展技术服务的边界

S



Security 安全技术

解决数字身份认证
实现有效风控
保护用户隐私
交易风险率低于千万分之一

I



IOT 物联网

智能互联
联通互联网与真实世界

C



Computing 计算技术

云技术降低技术和创新成本
大数据缓解信息不对称
有效甄别风险
让普惠金融可持续

区块链构建从“信息互联网”到“价值互联网”的桥梁



信息互联网

过去的20年通过“上网”
解决信息不对称的问题
互联网改变了人们的生活和交流方式



消费互联网

现在，通过“上云”
互联网从服务消费者的资源
进化成企业数字化转型的关键型支柱

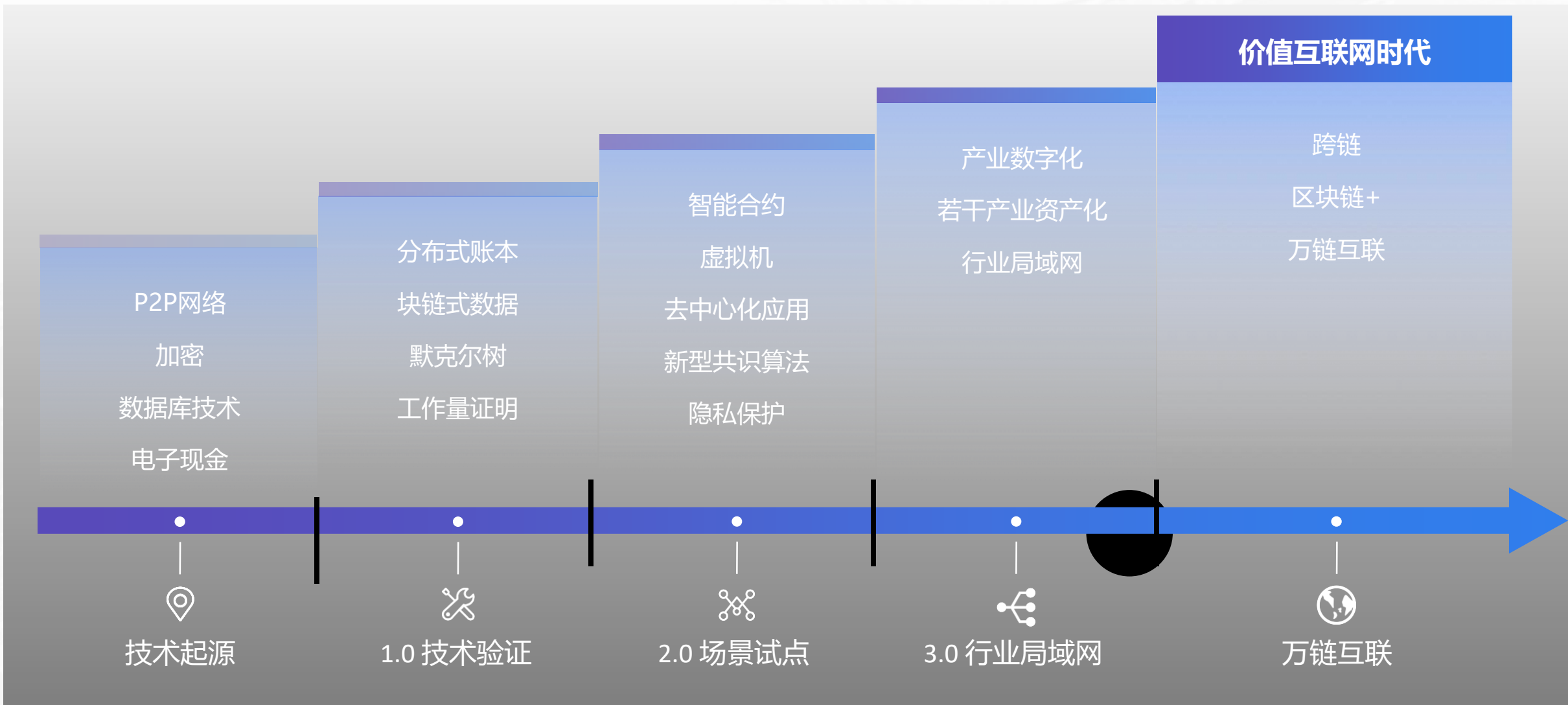


价值互联网

未来的20年通过“上链”
让更多的价值（资产）数字化
让人、设备、商业、社会协同创新

区块链技术的发展现状及展望

区块链并不是一项全新的技术，它是对多项成熟技术的一种创新型组合应用方式



区块链技术发展趋势

过去：公链与加密货币

以比特币和以太坊为代表的公链网络为主，加密货币无法锚定现实价值，导致与现有商业社会隔离

公链的底层技术体系面向无准入的C端网络节点，侧重开放与安全，导致交易性能低下

现在：联盟链与行业协作局域网

全球主流厂商进入区块链领域，推动技术突破和商业化场景落地。联盟链架构成为主流技术路线

聚焦研发高吞吐低延时大规模共识网络，开始构建在各行业内提升多方协作效率的价值链接局域网

将来：数字经济社会的价值互联网

随着各行业的数字化，行业协作局域网将通过跨链等技术完成万链互联，区块链将构建数字经济时代数据和资产可信流转的全球基础设施

AIOT技术推动物理世界与链上资产的锚定，进一步拓展价值互联网的边界

蚂蚁链构建下一代价值交换网络



蚂蚁链技术平台全景图



我们始终坚持解决真实的问题

2018年6月25日，在香港的区块链跨境汇款发布会上提出“三做、三不做”原则

“三做”：

用区块链来解决有社会价值的实际问题；
不断攻克区块链通向大规模实际应用的技术障碍；
与合作伙伴一起共建开放的区块链生态。

“三不做”：

不做没有任何真实价值的“空气币”；
不做违反法律法规的技术应用；
不做任何伤害用户数据安全和隐私的行为。

蚂蚁链业务现状

数字资产交易网络

数字资产交易市场

产品

资产
发行

资产
融资

资产
交易

资产
管理

...

供应链
金融

数字
物流

贸易
金融

数字
权益

50+
落地场景

行业协作网络

资产数字化应用

产品

联盟
构建

价值
锚定

注册
登记

资产
确权

...

财税
票据

可信
存证

数字
版权

溯源
商流

可信数
据网络

100万+
连接企业量

BaaS 平台网络

基础区块链平台

产品

单链版“套件”
(链所有权)

联盟公链“单
间”(链使用
权)

BaaS+增
值服务

开发
联盟链

数字
身份

...

1亿+
每日上链量



北京大學
PEKING UNIVERSITY

PART 二

蚂蚁链技术介绍

基础服务 ← 平台核心 ← 基础技术

网络规模指标

- 千计联盟网络
- 10亿月日均交易量
- 万亿计资产流转量

区块链基础服务

- BaaS云服务
- 数字身份服务
- 密钥管理服务
- 链外增值服务

BP网络核心

Layer 4 应用层

Layer 3 跨链交互层

Layer 2 链下通道层

Layer 1 交易共识层

Layer 0 网络路由层

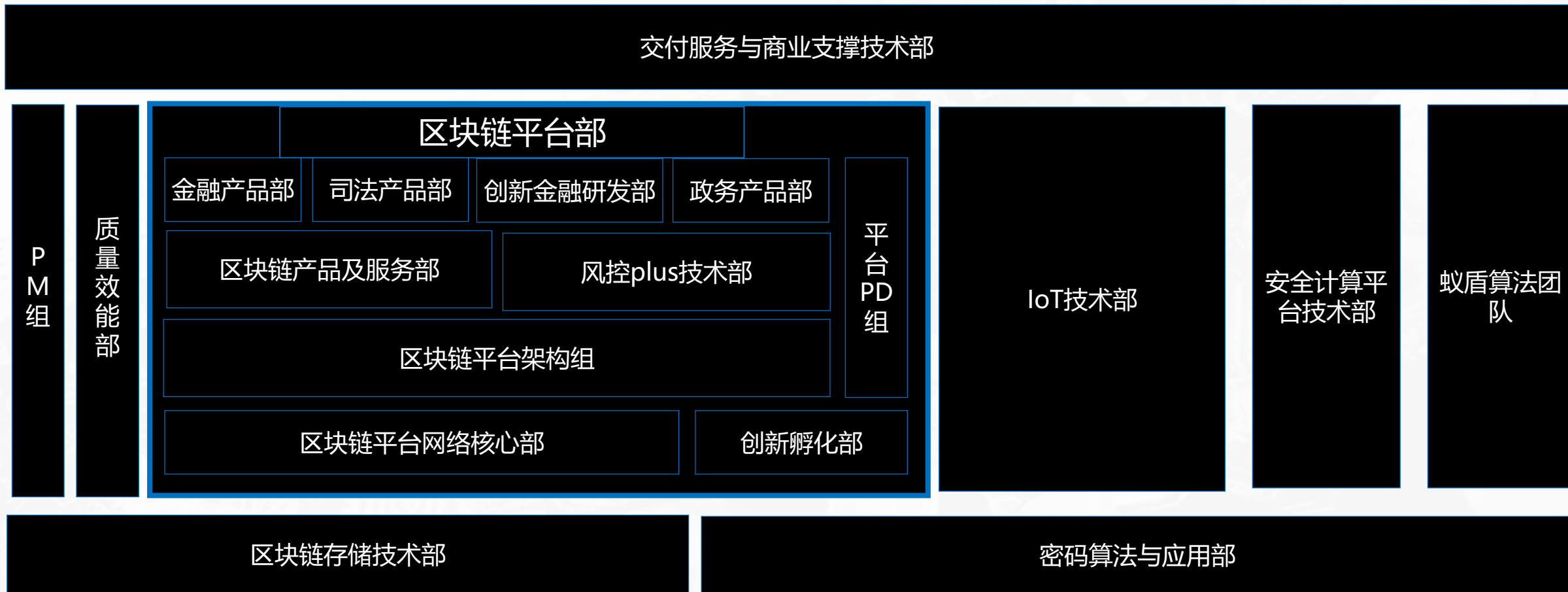
基础技术

- AIOT:**
扩展数字化资产物理边界
- 计算:**
面向智能合约新型计算芯片及云服务
- 存储:**
支持大规模账户的新型区块链存储服务
- 网络:**
广域网共识节点接入骨干网络
- 密码:**
隐私友好的信息使用与代数优化

核心技术指标

- 1. 自主创新（唯一性）**
2项面向区块链的新型计算硬件和网络服务。
1个金融级支付链路记账服务平台
- 2. 业界领先（领先性）**
全球最快的WASM合约执行引擎，业界第一个商业化的3D合约安全分析服务，隐私保护开发者友好性大幅提升。
- 3. 引领标准（影响性）**
核心技术推标准，如跨链IEEE标准，国家区块链安全标准，隐私保护ISO标准。

蚂蚁链技术兵力大盘架构



基础技术

跨链协议

可信硬件跨链验证技术 达到毫秒级延时及**10W PPS**性能。跨链协议已立项**IEEE国际标准**。

智能合约安全

智能合约自动化分析技术，提供新型**智能合约形式化验证和模糊测试**服务。与业界主流检测工具对比，在漏洞覆盖率、检测准确度和效率上有高达**3倍**提升

虚拟机

业界最快的智能合约**WASM虚拟机执行技术**，通过**JIT**技术取得接近原生代码的性能。单线程性能高达到业界同类产品**20倍**，同时支持多线程并行计算。

安全与隐私

代数优化的业界最快**PAILLIER99同态加密算法**。全球首个商用落地的基于可信硬件**区块链通用隐私保护方案**，国际数据库**顶级会议 'ACM SIGMOD 20'**上发表相应技术论文，拥有**80+**技术发明专利。面向**信任根完全自主可控**目标，创新研发阿里区块链安全计算硬件和国密安全芯片。

可验证存储

自研新区块链存储引擎，创新**高效安全的验证数据结构**设计，支持冷热数据分离和归档数据在线化查询，成本降低**50%**。

共识算法

原创带宽友好和可重启恢复的异步**BFT共识算法**，支持全球大规模部署。在上百节点场景中，**达到3W TPS和2s的确认时延**（对比：**Libra**仅为**1K TPS和10s确认时延**）

BTN网络

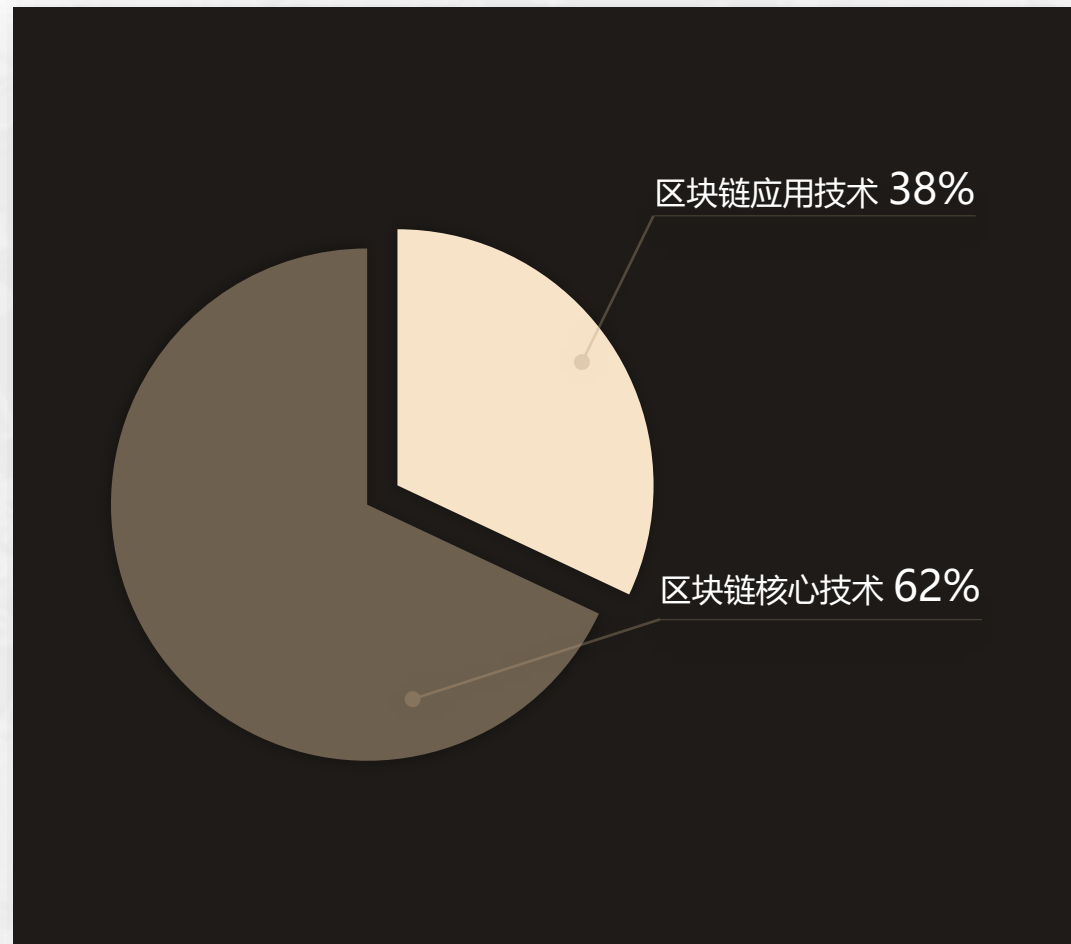
自研**区块链网络加速服务**，基于阿里POP网络**全球可接入**，在跨境多国共识网络部署条件下，区块链交易时延降低**20%以上**，节点带宽成本降低**30%以上**。

蚂蚁链支持蚂蚁自研区块链平台以及Hypeledger, Quorum等开源平台, 依据强大的研发能力, 蚂蚁链自研区块链平台具备以下差异化优势:

- **性能优势:** 蚂蚁链已经具备支撑“双十亿”的大规模应用能力。通过可拓展共识以及双层网络的设计, 辅以自研高效合约引擎, 在技术上蚂蚁链已经能够支撑10亿账户规模, 同时能够支撑每日10亿交易量。这让区块链技术在大规模商业场景中的应用成为可能。
- **密码学技术:** 区块链是生产关系的重构, 组织在协作的过程中关注商业隐私。蚂蚁链在隐私保护方面, 通过代数优化, 业已实现了六倍快于业界标准的同态加密算法。
- **存储技术:** 区块链连接的是数据, 随着大规模应用上链后, 必然带来存储的压力倍增, 蚂蚁链基于全新可验证存储数据结构设计, 在支持高效客户端交易验证能力中创新地采用数据压缩+冷热数据分离技术降低了至少一半的账本存储成本。
- **跨链技术:** 是区块链万链互联的基础, 蚂蚁链基于链下可信隐私计算平台, 提供具备动态扩容的可信计算集群处理能力, 使跨链平台具备10W PPS的跨链数据包处理能力, 遥遥领先业界。
- **隐私执行环境:** 由蚂蚁自主研发的区块链智能合约隐私安全处理器(基于FPGA实现), 在提供智能合约安全隐私执行环境的同时给用户可提供配置信任根的能力, 可实现Intel SGX技术的国有化替换。为践行普惠的理念, 蚂蚁集团已经完全开放区块链技术能力, 以BaaS(blockchain as a service)等的形式开放给客户与合作伙伴使用。通过结合区块链与云计算的特点和优势, 降低区块链应用开发门槛业务和开发成本, 携手生态共同探索区块链技术与商业应用的未来。

专注核心技术突破，领跑全球专利授权榜

排名	企业简称	国别/地区	全球区块链发明专利授权数量/件
1	阿里巴巴	中国	212
2	IBM	美国	136
3	Coinplug	韩国	107
4	Accenture	美国	49
5	美国银行	美国	49
6	MetAps	韩国	44
7	Ncchain	英国	43
8	MasterCard	美国	42
9	腾讯	中国	42
10	BlackGold coin	中国	41



数据来源：中国专利协会<2020年区块链全球授权专利报告>

蚂蚁链技术亮点示例：通用智能合约平台——MyChain

智能合约、数据存证、国密、大规模、高性能、高可用

定位&目标

- 面向**数据资产**：
多方各自拥有自己的隐私数据资产，通过蚂蚁链做灵活的数据资产登记、确权、交易、组合生成新数据资产，授权配置使用及计算。

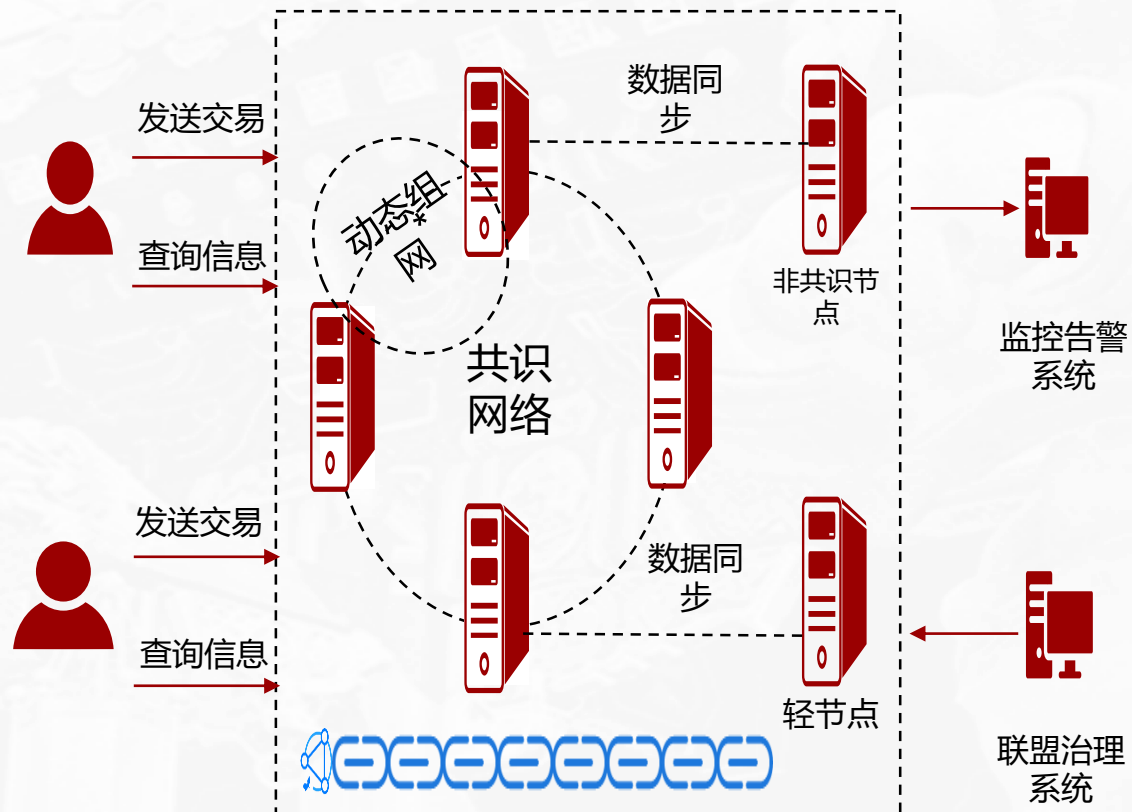
技术亮点

- 拜占庭容错：支持成员变更和满足BFT容错的可适应性共识算法
- 并行性执行：符合串行执行语义的交易并行执行技术
- 大规模存储：支持10亿账户规模的可验证数据读写能力
- 国密安全性：符合国密资质的密码学算法与密钥保护技术
- 强隐私保护：提供ZKP、TEE和物理隔离的多重隐私保护手段

场景与性能

- 156条——已上线服务于金融、司法、物流、医疗、政务、公益、生活等领域50多场景
- 1亿账户规模下1亿日交易吞吐能力

思想自由 兼容并包



蚂蚁链技术亮点示例：大账户规模高交易吞吐资产交易平台——ALDABA

大规模、高吞吐、数字资产、安全、全球部署

定位&目标

- 安全、可靠、大规模、高性能、全球部署的资产交易平台
- 10亿级账户规模、10亿级日交易处理能力；水平扩展能力
- 技术原创性和领先性

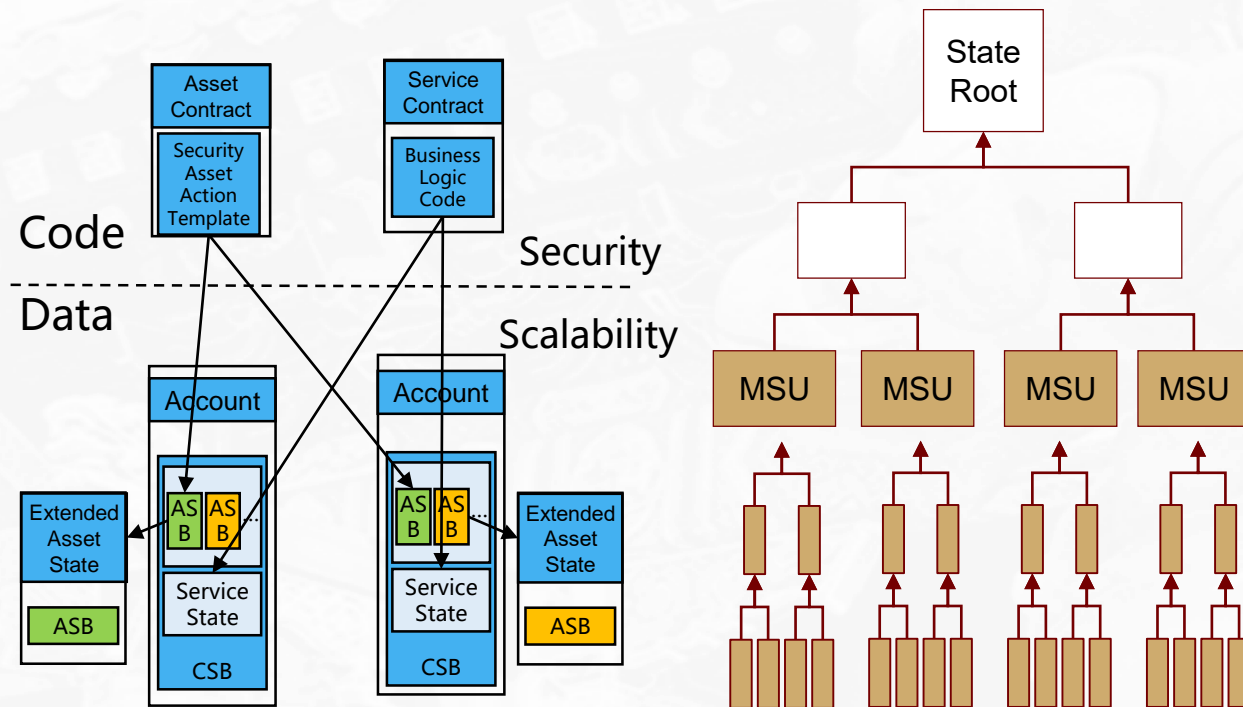
技术亮点

- **安全性**：合约执行运行时底层资产状态变更跟踪和完整性保障
- **高性能**：分片水平扩展；LOAD, COMPUTE, STORE 高效合约执行
- **易用性**：服务合约和资产合约解耦，隐私和明文资产操作方式相同
- **可升级**：协议层支持节点(链)和合约(应用)升级
- **创新性**：新世界状态模型、状态空间分片、内置的升级支持能力

场景与性能

- 账务系统、支付，简单及多维复杂资产应用：集分宝、跨境汇款
- 10亿账户下3万TPS

思想自由 兼容并包



合约 code 和 data 分离
对象数据 data 横向可扩展
资产数据操作安全性

基于世界状态空间切分
MSU实现节点间分片异构

蚂蚁链技术亮点示例：链上通用高效 (TEE)隐私保护方案——MYCloak

关键字：隐私保护、数据授权、通用、高性能、形式化安全

定位&目标

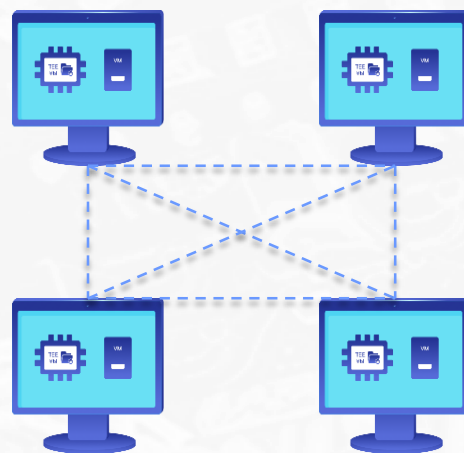
- 蚂蚁区块链平台通用、高性能、高安全的区块链链上数据**隐私保护**以及**隐私数据授权**解决方案，保障交易数据端到端、全生命周期的隐私安全

技术亮点

- **通用性**：适用于任意智能合约应用场景的数据隐私保护需求
- **高性能**：基本保持明文计算性能 (**0**成本的隐私保护)
- **灵活性**：隐私模型和数据授权机制灵活可编程定义
- **安全性**：隐私安全协议通过形式化验证
- **兼容性**：已有平台明文数据**0**成本升级至具有隐私保护的平台

场景与性能

- 已上线业务场景：**双链通、ABS、IoT仓单融资**等
- 支持基于AoT的原生合约执行能力和并行执行能力



- 硬件全链路交易隐私安全
- 明文、密文结果混合打包成块共识



隐私交易

- 加密提交
- 在TEE*中解密执行
- 结果加密输出



非隐私交易

- 在TEE外执行

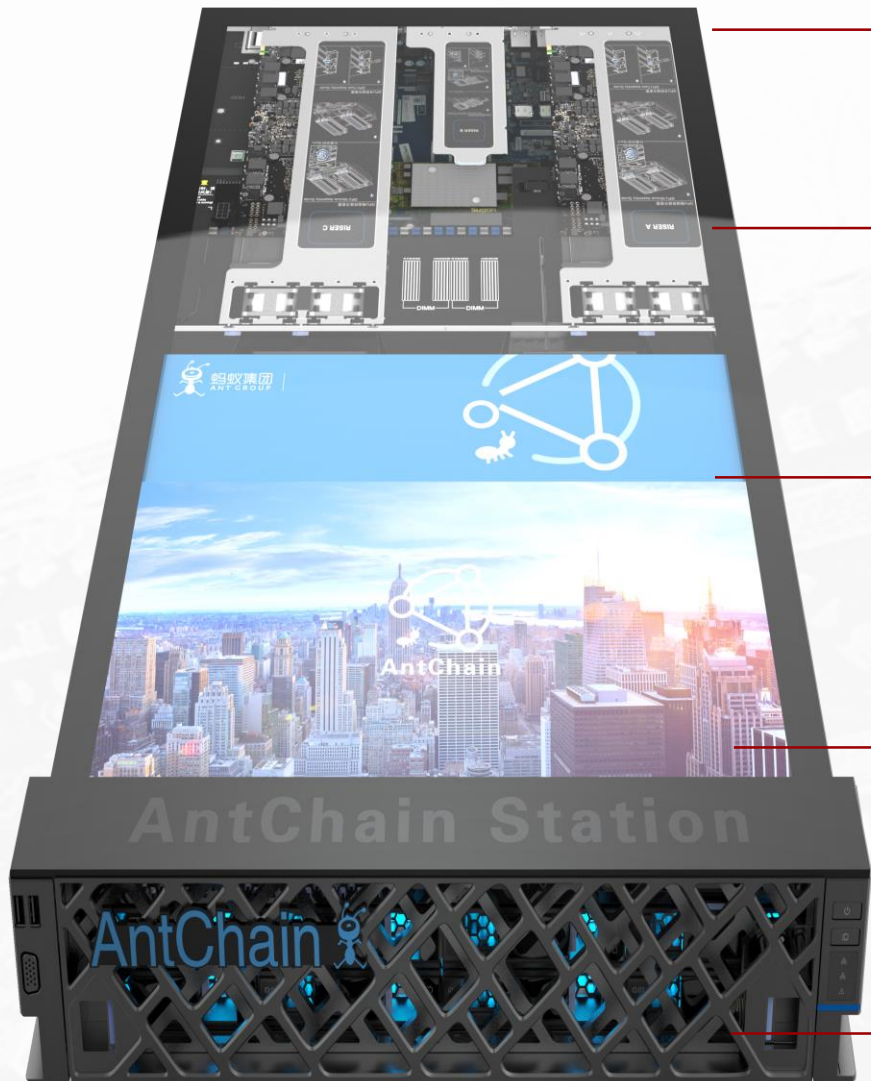
***可信执行环境(TEE)**：基于CPU硬件的安全、可信、隔离的执行环境，确保可信区内代码和数据的完整性和机密性。具有性能高，安全可审计和可认证的特性。

技术先进性

- 全球首个商用落地区块链硬件隐私保护技术
- 技术论文发表于国际数据库顶会(ACM SIGMOD' 20)
- 自主可控的硬件隐私合约专用处理器(0信任根依赖、替代SGX)集成在**蚂蚁区块链一体机**中

蚂蚁链技术亮点示例：软硬协同一体机

携手阿里和平头哥团队



国际/国产化双配置可选底座，携手平头哥联合推出 (X86+CENTOS, ARM+UOS)



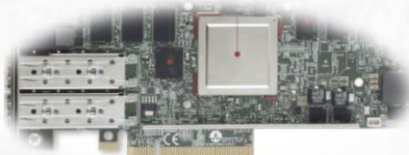
安全计算硬件

- 智能合约隐私计算硬件环境
- 自主可配置信任根 (不依赖硬件厂商)



密码卡

- 硬件执行加密算法，提升5-10倍性能
- 获得国密最高 (三级) 资质



网络共识加速器：软硬协同共识算法设计，降低**40%**共识延迟

蚂蚁链：提供从底层技术平台到上层场景解决方案全栈服务，开箱即用

蚂蚁链技术亮点示例：蚂蚁链BaaS



帮助用户应用上链

BaaS Core: 核心功能层, 保障区块链服务稳定可用, 提供用户访问和管理区块链的服务和工具。

BaaS Plus: 三方服务集成, 帮助用户更快速的利用已有的服务与区块链技术结合。

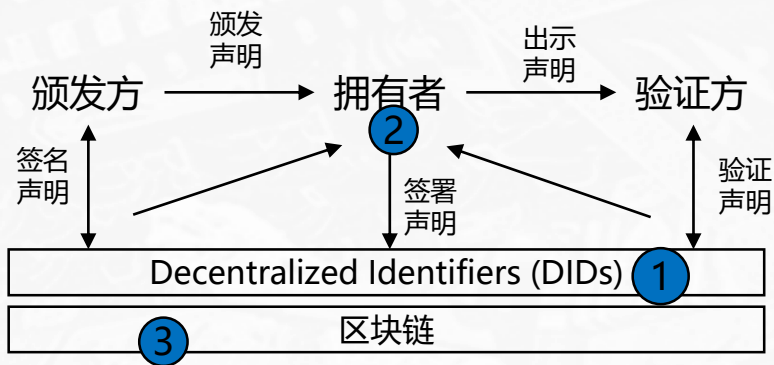
中间件层: 链上和链下组件扩展, 一站式的上链保障, 链基础能力创新。

开放联盟链: 为中小企业、开发者设计的区块链, 价格低, 无需搭建, **3步上链**。
 周期: 周->分钟
 成本: 万->百
 起步成本: -100倍。

区块链的本质是数字资产可信流转

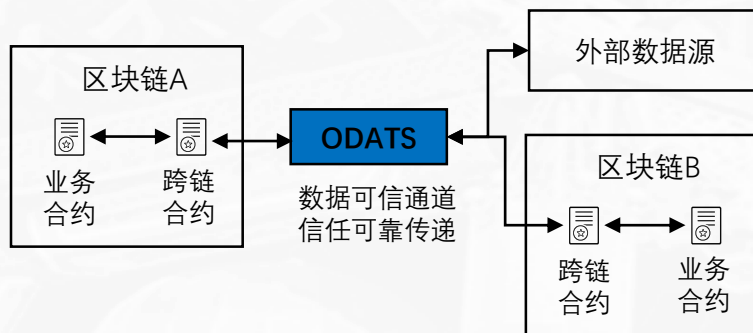
数据主权

区块链数字身份（所有者/使用者）DIS：身份自主控制，确定数据所有权和使用权。



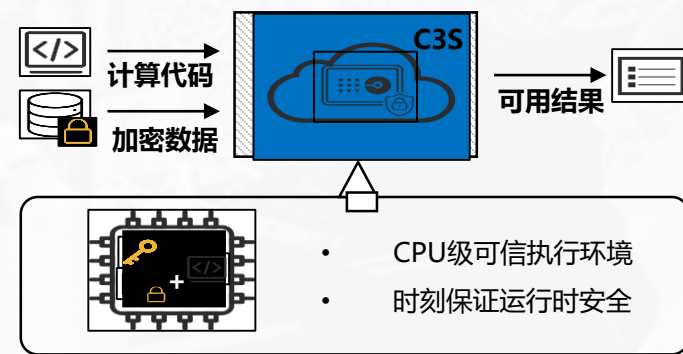
数据连接

跨链和可信数据源ODATS：连接链与链之间，链上与链外之间的可信数据。

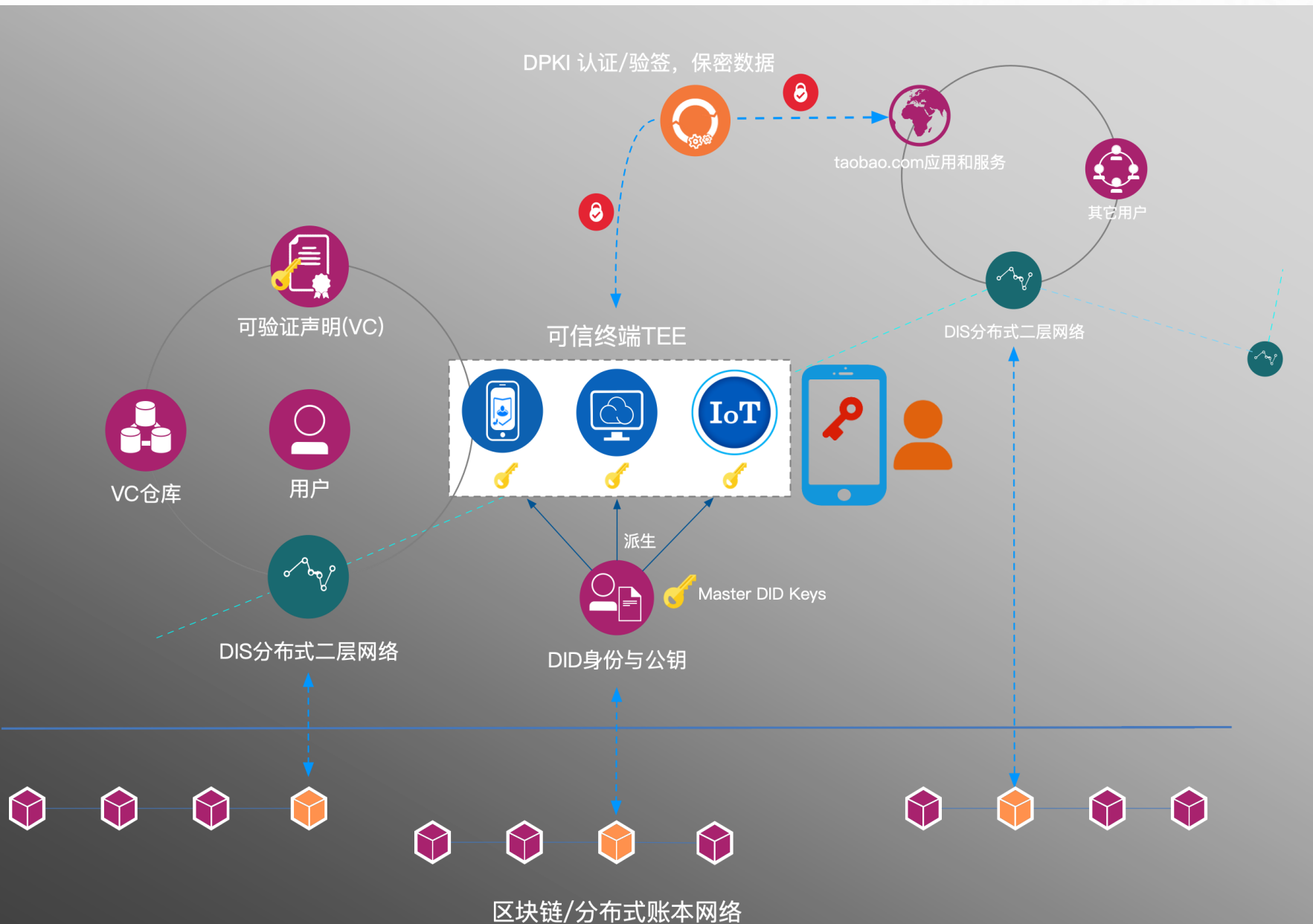


数据流转

数据隐私计算（流转）C3S：全链路未暴露原始数据。计算结果可信、可用。



蚂蚁链技术亮点示例：蚂蚁链数字身份



● DIS/DPKI自主权管理

现实身份与链上身份对应，实现具有自主权签名认证体系，直接证明身份意愿，我的数据授权。

● 端到端隐私保护

终端TEE配合链上TEE，通过DPKI体系实现端到端隐私保护、可信计算。

● 数据最小化可见

保证可验证声明不被篡改的前提，支持基于默克尔证明、零知识证明技术的最小化信息披露，保护隐私。

● 二层网络高可用

通过二层网络将上层的请求进行归并，并通过账本分片，合约分片等技术，实现在海量账户的情况下对外服务高可用。

● 派生与归并

为了保护身份行为轨迹，用户可以通过一个主数字身份去派生出全新的数字身份，用于不同的场景中；而需要时用户可以证明不同数字身份归属于同一个实体。



北京大学
PEKING UNIVERSITY

PART 三

蚂蚁链产业实践分享

蚂蚁链走入大众 落地场景超过50个

公益时间上链
兑换养老服务

4亿件海外商品上链
有了「正品身份证」

500万「卖家秀」上链
版权有保障

小企业身份上链
全球贸易身份认证时间缩短5分钟

3万卖家账款上链
在「双链通」获得贷款

超2亿资金募捐上链,
善款来源可追溯

医疗票据上链
1分钟完成异地报销

发票上链
3秒开发票

社保上链
1分钟异地提取公积金

菲律宾、马来西亚等国实现
区块链跨境转账、实时到账

证据上链
数字证据成为判决依据

处方上链
用户坐等送药上门

合同上链
手机签合同只要1分钟

注册资本30元的小企业,
靠和上游公司合作
3分钟贷到周转金

农产品上链
全链路保真, 农民收成更高

房屋租赁上链
管理真实房源, 消灭黑中介

相互宝上链
确保互助过程透明不可改

地铁票上链
用户一码畅行13城

EMS上链
客服投诉降低60%

蚂蚁链 链接产业价值

政务服务



江西省政府



杭州市政府

港口贸易



招商局港口



中国远洋海运

融资租赁



英特尔



戴尔、惠普、联想、民盛

蚂蚁链+监管平台：加强监管，保障安全，维护权益

食品药品溯源



危化品溯源



工程质量溯源



蚂蚁区块链溯源

支付宝扫码

溯源查真伪

	溯源码	
	890000	
	000001	
	345600	



政府

企业信息
资质信息
产品信息
产量信息



加工企业

企业名称
加工过程
生产批次
包装需求



质量监督局

包装审批
质量检测
产量匹配
包装激活



物流

公司名称
启运地
启运时间
运输时间



仓储

产品明细
入仓时间
调拨情况
出仓时间



配送

订单关联
启运地
配送公司
发货时间



消费者

确认收货
验真查询
查询时间
查询地点



蚂蚁链+司法：让数据成为证据，让合同成为契约

司法区块链平台

基于全球领先的区块链技术构建内外网一体的开放式司法链平台，提供司法可信服务，实现司法活动中的电子数据可核验、可追溯，助力司法改革，全面提升法院司法能力、司法效能和司法公信力。



- 数据上链存证
- 司法链数据核验

将各类数据上链，诉讼中可作为电子数据证据
基于已上链数据，验证电子数据证据效力



重磅！杭州颠覆购房签约方式！“云签约”领先全国

钱江晚报 2020-04-15 08:45

4月13日，住建部发布《关于提升房屋网签备案服务效能的意见》，其中提到，要推进“互联网+网签”，积极鼓励推行“互联网大厅”模式，实现房屋网签备案掌上办理、不见面办理；在签章确认环节时，有条件的城市，可以采用电子签名（签章）技术，在网签备案系统中予以确认。

而杭州又一次走在了全国前列——近日，杭州市住保房管局正式推出买房卖房“云签约”，买卖双方无需到签约现场，通过手机



最高法司法链

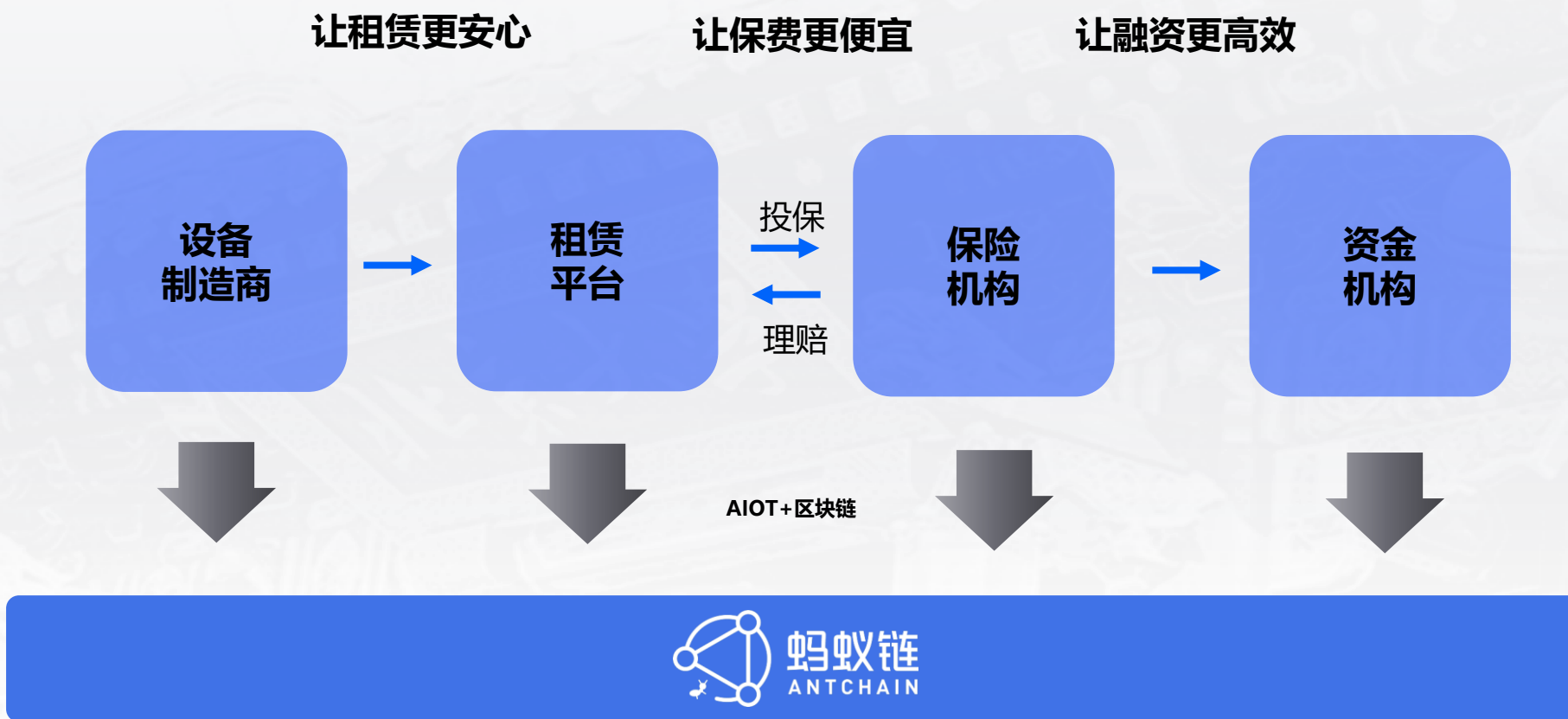
联合最高法的四级多省市共上千家法院，

完成超90亿条数据完成链上固证，32节点数。

全流程记录+全链路可信+全节点见证



蚂蚁链+租赁：生态聚合，创造新生业态





北京大学
PEKING UNIVERSITY

PART 四

蚂蚁链开发者实验室

它是技术开放阵地

蚂蚁链开发者实验室是蚂蚁链首个面向开发者搭建的基础技术环境，包含覆盖前、后端的区块链框架、工具和服务，以及从入门到进阶的区块链实训教程，开发者可以根据教程自主调用技术模块进行应用开发。

该平台的主要特点是“**低门槛**”，即使是初学者，也可以快速搭建一个区块链应用。开发者实验室中集成了蚂蚁链多个支撑实际业务场景的企业级区块链中间件，可以实现“开箱即用”。

早期，很多对性能、安全等条件没那么高要求的应用主要依托智能合约做计算，但是随着企业级场景越来越丰富，就需要一整套的技术基础服务做支撑，比如云服务集成技术、中间件、合约安全、链上链下一体化等。这也对开发者提出了更全面的要求。

蚂蚁链以区块链平台技术为起点，形成了一套全栈的技术体系，涵盖平台技术、扩展技术、融合技术和应用技术，**可满足开发者不同要求的“用链”需求**。

如支撑多款生产级的区块链应用框架ChainStack、DApp研发平台Workbench、专业集成开发环境Antchain Studio、前端开发框架Myfish、在线合约开发工具CloudIDE等。

开发者只需登录就可以使用蚂蚁链“同款”开发技术。

它是开发者成长中心

开发者实验室同时也是体系化的区块链技术学习中心，开发者可以在实验室一站式学习、动手实训和完成技术能力认证。

学习平台提供**100+在线学习课程**，既可以完整学习区块链基础知识，了解底层原理及核心实现，又可以深入学习蚂蚁链技术的企业级特性和原理。

实训平台提供手把手式的区块链编程实训教程，让开发者可以step by step入门开发区块链，进阶使用区块链。

对开发者入门友好也是一大特点，如实训课程中，solidity语言入门开发教学使用了“蚂蚁搬家”的趣味教学供学习语法。

实战

为了把小蚂蚁保存在我们的合约里，并且能够让其它合约看到这些蚂蚁，我们需要一个公共数组。

1. 创建一个数据类型为 Ant 的结构体数组，用 public 修饰，命名为：ants.

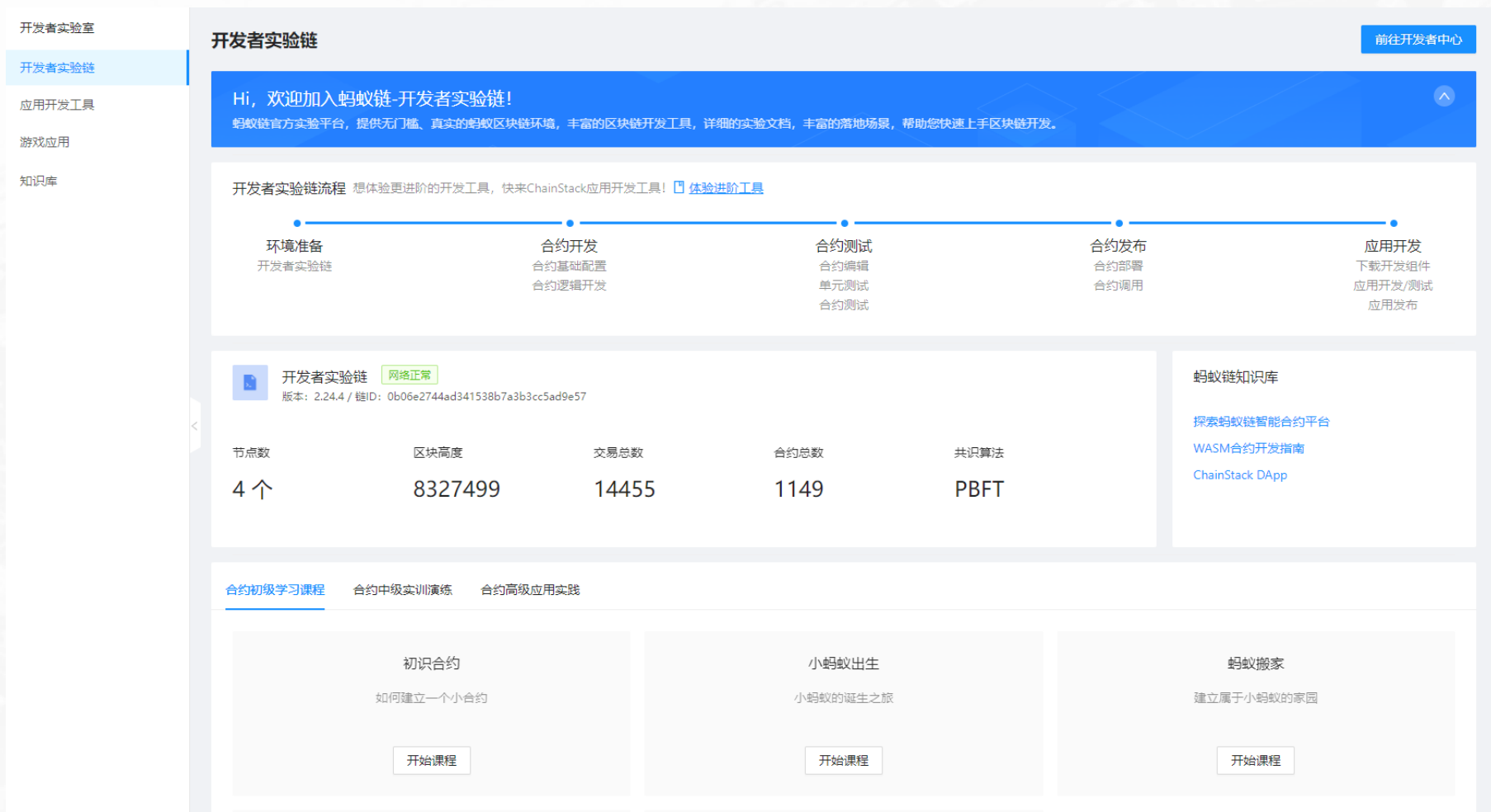
在以下编辑框中编写程序

```
1  pragma solidity ^0.4.20;
2
3  contract AntFamily {
4
5      uint dnaDigits = 12;
6      uint dnaModulus = 10 ** dnaDigits;
7
8      struct Ant {
9          string name;
10         uint dna;
11     }
12
13     // 在这里写
14
15     function createAnt(string _name, uint _dna) {
16
```


开发者实验室有什么？

1) 开发者实验链

开发者实验链是基于蚂蚁链合约平台的专业版联盟链，利用它开发者可以免费体验真实的链上开发环境，实践区块链技术，开发测试DApp。



The screenshot shows the 'Developer Lab' interface. On the left is a navigation menu with 'Developer Lab' selected. The main content area has a blue header with a welcome message and a 'Go to Developer Center' button. Below is a 'Developer Lab Process' flowchart with six steps: Environment Preparation, Contract Development, Contract Testing, Contract Deployment, and Application Development. A status bar shows 'Developer Lab' is online with version 2.24.4. A table displays network statistics: 4 nodes, 8327499 block height, 14455 transactions, 1149 contracts, and PBFT consensus. A 'Knowledge Base' section lists links for exploring the platform, WASM contract development, and ChainStack DApp. At the bottom, there are three course cards: 'Introduction to Contracts', 'AntChain Birth', and 'AntChain Migration', each with a 'Start Course' button.

开发者实验室

开发者实验链

应用开发工具

游戏应用

知识库

开发者实验链

前往开发者中心

Hi, 欢迎加入蚂蚁链-开发者实验链!

蚂蚁链官方实验平台, 提供无门槛、真实的蚂蚁区块链环境, 丰富的区块链开发工具, 详细的实验文档, 丰富的落地场景, 帮助您快速上手区块链开发。

开发者实验链流程 想体验更进阶的开发工具, 快来ChainStack应用开发工具! [体验进阶工具](#)

- 环境准备
开发者实验链
- 合约开发
合约基础配置
合约逻辑开发
- 合约测试
合约编辑
单元测试
合约测试
- 合约发布
合约部署
合约调用
- 应用开发
下载开发组件
应用开发/测试
应用发布

开发者实验链 网络正常
版本: 2.24.4 / 链ID: 0b06e2744ad341538b7a3b3cc5ad9e57

节点数	区块高度	交易总数	合约总数	共识算法
4 个	8327499	14455	1149	PBFT

蚂蚁链知识库

- [探索蚂蚁链智能合约平台](#)
- [WASM合约开发指南](#)
- [ChainStack DApp](#)

[合约初级学习课程](#) [合约中级实训演练](#) [合约高级应用实践](#)

- 初识合约
如何建立一个小程序
[开始课程](#)
- 小蚂蚁出生
小蚂蚁的诞生之旅
[开始课程](#)
- 蚂蚁搬家
建立属于小蚂蚁的家园
[开始课程](#)

2) 应用框架

- **【ChainStack框架】** 企业级区块链应用研发框架，涵盖智能合约框架与类库、后端应用框架、应用原型等，基于它可以快速构建企业级区块链应用；
- **【MyFish框架】** 一款面向前端开发者的蚂蚁链DApp研发框架，支持TypeScript和Solidity智能合约开发，支持前端应用快速集成；

3) 开发工具

- **【CloudIDE】** 一款在线合约开发IDE，为智能合约开发提供简单、高效的集成环境。
- **【Antchain Studio】** 一款本地专业IDE，蚂蚁链生态应用开发的集成开发环境，集成蚂蚁链全栈的合约开发工具链和开发者服务，可以进行更专业的企业级应用构建。

4) 中间件服务

- **【DApp Workbench】** 全栈dapp研发综合服务平台，集成各类应用中间件与合约服务，支持快速构建一个集成框架和中间件的企业级DApp。
- **【交易网关/数据网关】** 支持低门槛、高性能、高可靠的企业级区块链接入服务，支持准实时、高可靠的区块链消息事件及数据推送服务。
- **【合约治理服务】** 支持智能合约运维治理服务，多方可联合治理合约，也提供合约部署升级、合约业务参数变更等运维能力，让合约运维更简单。
- **【应用速搭平台】** 提供可视化编程合约，只需要配置少量代码就能完成开发测试，简单应用可以实现“0代码”一秒创建、3秒上链。
- **【合约审计】** 合约代码自动化检测服务，让漏洞无所遁形。

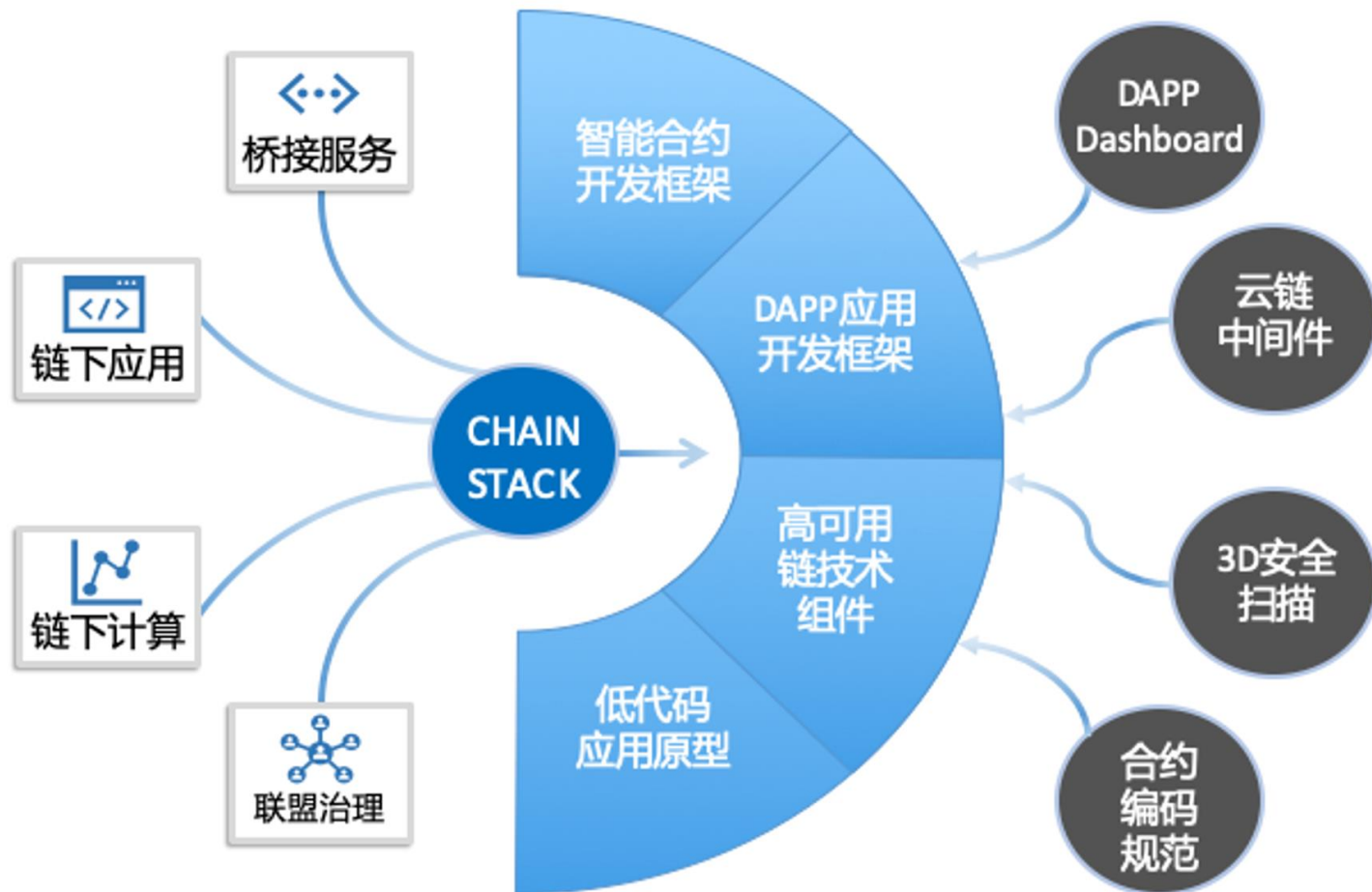
ChainStack概览

企业级区块链应用研发框架，涵盖智能合约框架与类库、后端应用框架、应用原型等，基于它可以快速构建企业级区块链应用。



ChainStack 经过对蚂蚁链内部业务的最佳实践进行挖掘、实践分享、组件沉淀探讨，从蚂蚁内部的业务线上区块链应用实现上提炼总结，形成一张蚂蚁链之上的区块链应用技术图谱。

基于该技术图谱，ChainStack 从框架与服务角度出发，打造服务于区块链应用的框架。

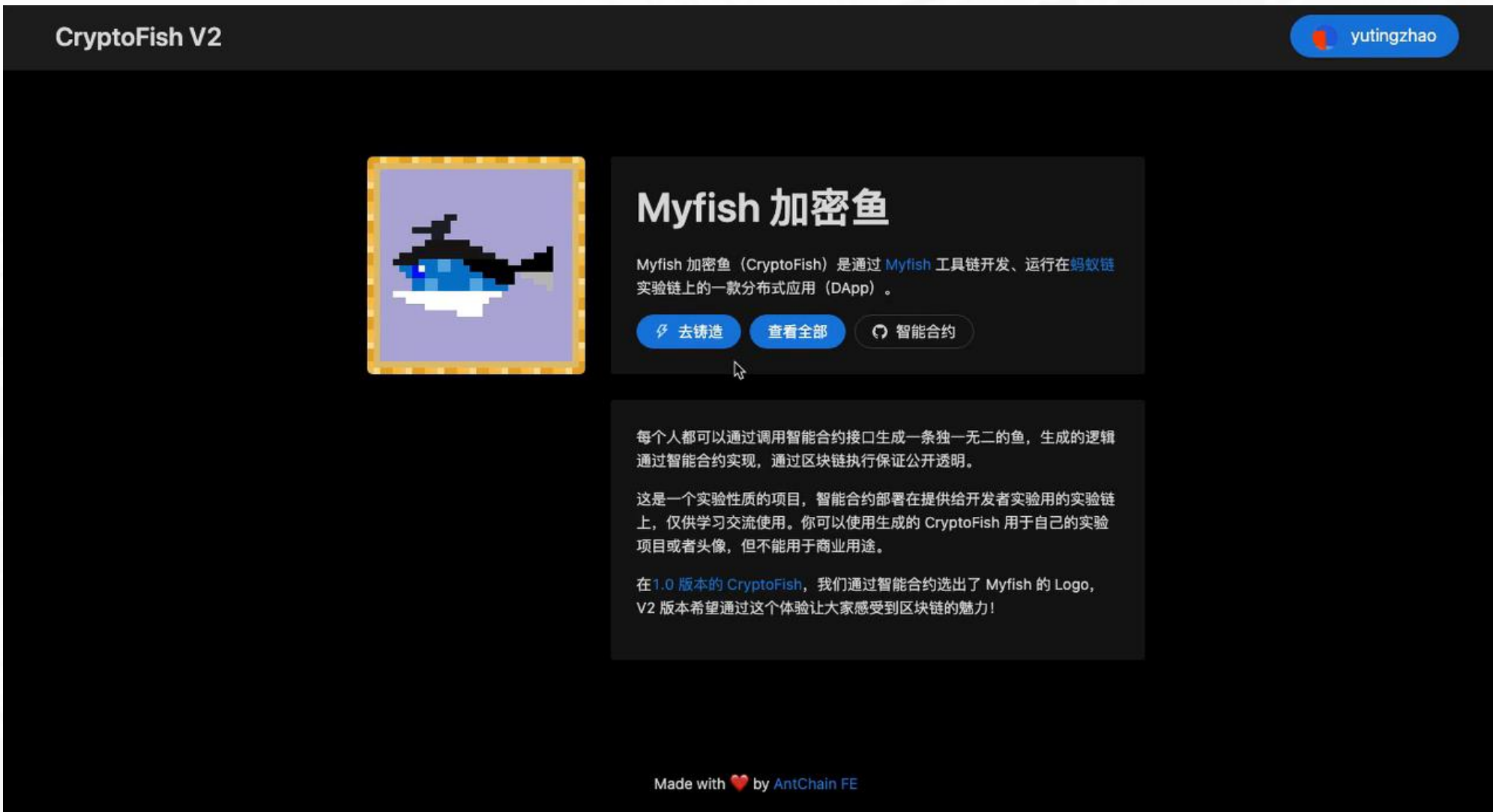


- **一套智能合约框架**，提供具备快速研发、组件复用、联动集成设施的智能合约框架，使得基于 ChainStack 的智能合约具备全面的生产化要素。
- **一套可复用的应用组件**，提供一站式集成区块链、智能合约、中间件的 DAPP 应用研发框架，以及配套多种解决方案的区块链应用原型，支持业务快速搭建 DAPP 应用。
- **一套完整的 DApp 应用工具链**，涵盖开发环境自动配置、应用/合约代码自动生成、IDE 适配等，覆盖 DApp 应用的整个生命周期。

- Myfish 是一个轻量级的 DApp 研发框架。面向前端开发者提供包括 Solidity / TypeScript 智能合约研发、纯前端 dapp 开发、蚂蚁链 JSSDK 等能力，支持快速研发一个纯前端的 DApp。
- Myfish 提供了一个命令行工具，你可以用它来初始化代码、编译智能合约、部署智能合约、查看链数据等。

MyFish——在线体验DApp

我们基于 MyFish 搭建了一个纯前端的 DApp, CryptoFish V2。
通过该 DApp 你可以通过智能合约随机创建一个独一无二的 CryptoFish 头像。该 DApp 源代码开放。



CryptoFish V2

yutingzhao

Myfish 加密鱼

Myfish 加密鱼 (CryptoFish) 是通过 [Myfish](#) 工具链开发、运行在 [蚂蚁链](#) 实验链上的一款分布式应用 (DApp)。

[去铸造](#) [查看全部](#) [智能合约](#)

每个人都可以调用智能合约接口生成一条独一无二的鱼，生成的逻辑通过智能合约实现，通过区块链执行保证公开透明。

这是一个实验性质的项目，智能合约部署在提供给开发者实验用的实验链上，仅供学习交流使用。你可以使用生成的 CryptoFish 用于自己的实验项目或者头像，但不能用于商业用途。

在 1.0 版本的 [CryptoFish](#)，我们通过智能合约选出了 Myfish 的 Logo，V2 版本希望通过这个体验让大家感受到区块链的魅力！

Made with ❤️ by AntChain FE

Cloud IDE 合约开发环境是 BaaS 合约平台提供的在线合约开发工具。

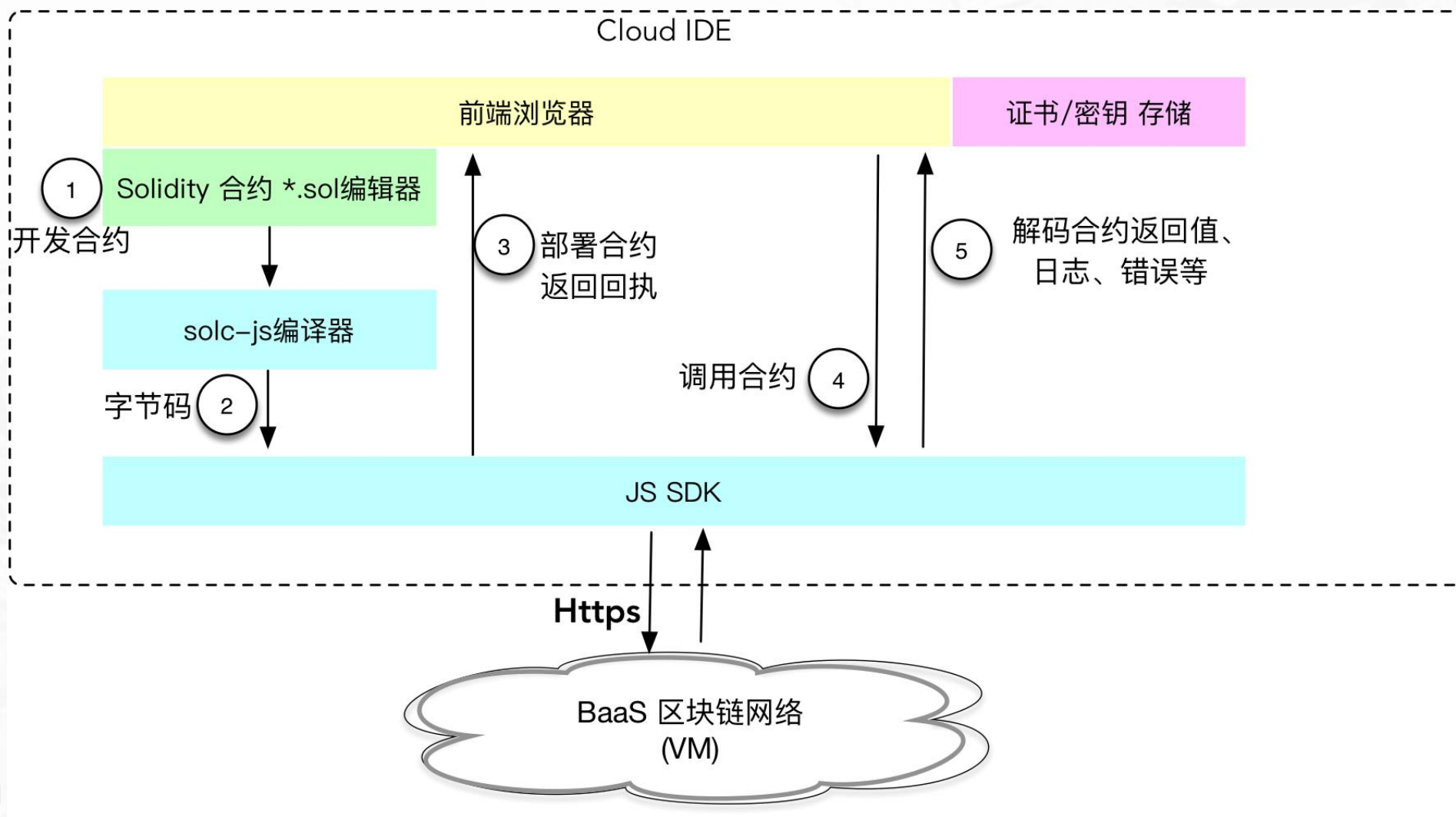
核心功能

Cloud IDE 为智能合约开发提供简单、高效的集成环境，并提供以下核心功能：

- **合约工程管理**：管理工程目录和文件，提供文件编辑自动保存功能，保存合约到 BaaS 合约管理。
- **合约编辑与编译**：展示编译结果字节码和接口说明（ABI）。
- **合约的部署和调用**：提供默认合约体验链环境和测试账户，用来部署和调用合约。
- **自动编解码**：解析合约方法的返回值、事件日志等，辅助合约测试。
- **合约单步调试**：当前支持 Solidity 合约的单步调试功能，可单步执行，查看临时变量、合约存储、堆栈、内存等信息来深度调试合约。
- **合约单元测试**：当前支持 Solidity 合约的单元测试，可直接使用 Solidity 语言编写单元测试用例，一键执行所有测试用例。
- **合约分析**：在编译合约的同时对 Solidity 合约进行静态扫描分析，给出分析结果参考。

Cloud IDE基本框架

Cloud IDE 是一个典型的去中心化应用 (Dapp)，可通过 JavaScript SDK 直接与区块链平台通信，进行合约部署和调用。这样设计的优势在于，通过 IDE 您可以连接任何目标的蚂蚁区块链平台环境。



使用 Cloud IDE 开发合约的过程主要分为以下 5 步：

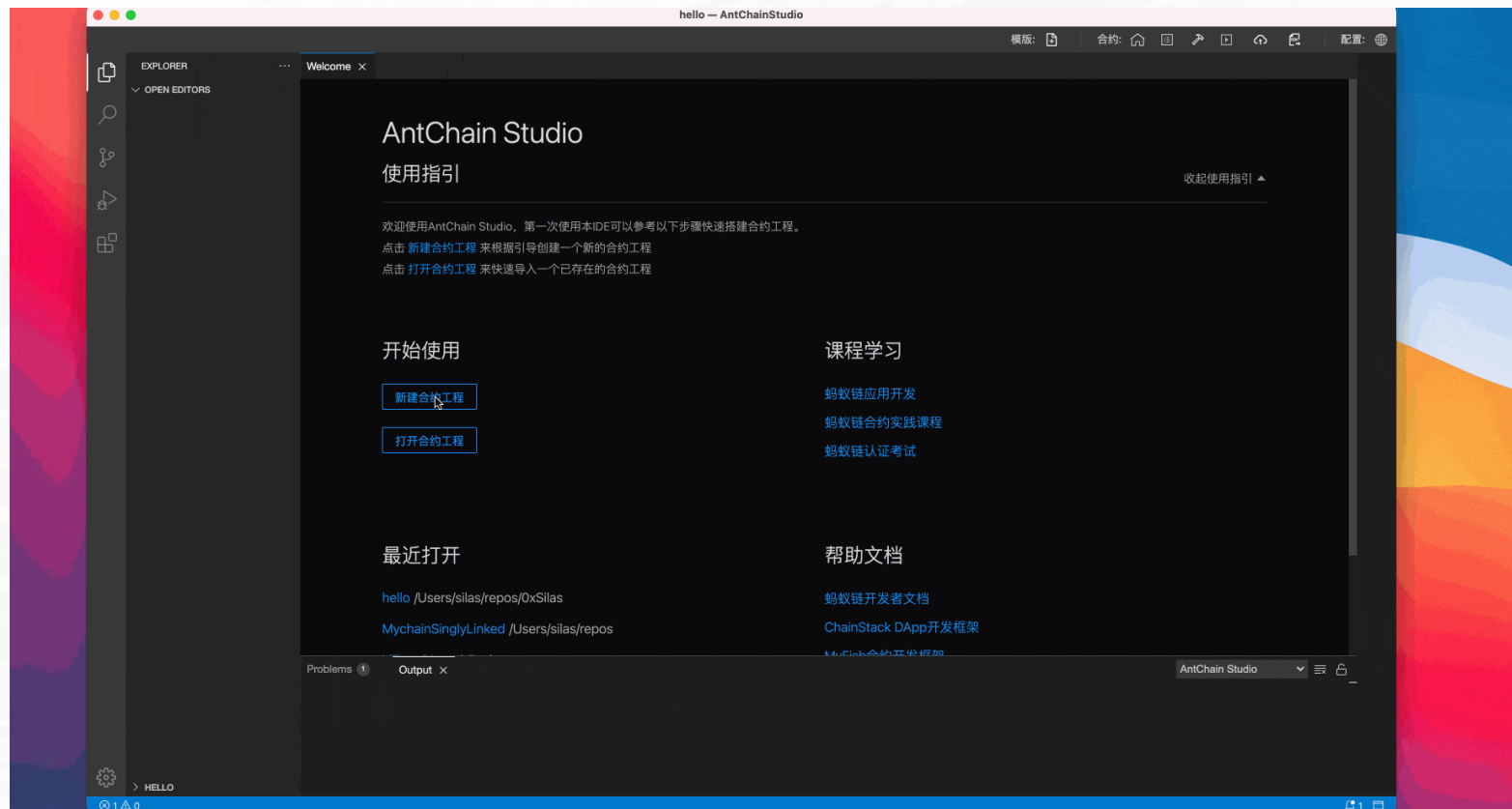
- 1.选择目标合约链：** 在环境配置中，您可以选择已有的公共合约体验链，该体验链会配备一个默认测试账户，可以用来部署、调用合约。如果此测试账户不能满足需求，您也可以使用通过 SDK 创建的账户。在环境配置中，您也可以选择自定义的合约链环境。
- 2.编译合约：** 开发合约过程中，可以随时对已完成的代码进行编译。编译后会生成字节码以及合约接口说明。
- 3.部署合约：** 在合约编译通过，得到字节码和ABI后，在 Cloud IDE 中单击部署合约将合约部署到目标体验链进行测试。部署成功后，可以看到合约的所有方法列表。
- 4.升级合约（可选）：** 对于正式发布后的合约，可以根据需要在 Cloud IDE 中进行合约升级。
- 5.调用合约：** 合约部署成功后，Cloud IDE 会列出合约中所有的 public 方法以及 public 类型的状态变量。单击目标方法右侧的调用合约进行调用。
- 6.调试合约：** 在调用合约过程中，可以根据出错时的错误码信息调试。也可以通过添加 Event 事件，根据日志进行调试。

- **合约单步调试**: 合约调试模块增加了一个虚拟机模拟器, 可以模拟运行 Solidity 合约并给出执行结果。进行单步调试, 可以查看运行时的局部变量、合约存储状态、堆栈、内存等关键信息进行深入调试。
- **合约单元测试**: 为方便使用, 在 IDE 单元测试模块中单击 添加测试文件 可以快速生成一个单元测试用例, 并且生成后的测试用例可以直接运行。
- **合约安全检测**: 合约安全检测平台通过 CloudIDE 的插件方式集成到 IDE 功能中, 开发者在使用 CloudIDE 开发和部署智能合约时, 可以创建检测任务, 对工程中的合约进行安全检测。检测结果中提供合约的相关统计信息和安全问题, 辅助和指导开发者更好的编写安全规范的智能合约。
- **使用 Cloud IDE 开发 TEE 硬件隐私合约链**: 在 Cloud IDE 中进行环境配置时, 可选择连接 TEE 硬件隐私合约链环境。启用 TEE 加密后, 提交的交易都是加密交易, 包括部署合约、调用合约。

Antchain Studio概览

AntChain Studio 是一款面向区块链智能合约研发，易用低门槛、多语言、快速研发、一体化集成的智能合约 IDE，让 dapp 开发更快更友好更专业！

AntChain Studio 提供了一站式集成开发环境，包括开发环境**自动配置**、应用/合约代码**自动生成**、合约**编译部署调试**等。



• 快速智能合约研发

配套专业智能合约研发课程与**可视化合约操作界面**，无需任何一行代码，新手用户也能**快速接入区块链网络**，一键创建智能合约工程、部署上链并可视化调试，实现用户快速研发、部署、调试智能合约。

• 一站式集成开发环境

一体化集成蚂蚁链区块链研发工具，涵盖 MYCDT 编译工具与 ChainStack 企业级研发框架，提供具备**低门槛、简单易用、快速研发**的智能合约研发框架以及配套多种解决方案的区块链应用原型，支持业务快速搭建 DAPP 应用。

• 一套企业级研发工具

依托蚂蚁链平台开放行业领先的金融科技区块链服务，提供开发环境自动配置、应用/合约代码自动生成、多语言合约编写，**一键编译部署**，快速调试的智能合约开发服务，覆盖 DApp 应用的整个生命周期，加速用户智能开发、测试。

• 可视化调试

区别于传统智能合约研发方式，AntChain Studio 为用户提供**可视化调试界面**，只需要选择想要调试的合约接口并填写测试数据，即可自动保存测试用例、一键运行并展示测试结果。

• 一套高效友好开发框架

一套专门为区块链应用 dapp 打造的高效友好研发框架，涵盖智能合约框架与类库、dapp 后端应用框架、spring 集成、应用原型等，适用于企业级区块链应用平台研发。

AntChain Studio 核心功能



快速智能合约研发

一站式集成开发环境与可视化操作界面, 帮助用户低门槛快速研发



可视化合约接口

可视化展示合约接口列表与合约事件列表, 支持企业级工具使用



编译部署

企业级高效编译工具链, 支持多语言合约, 一键部署上链



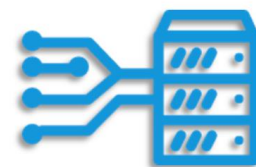
模板市场

展示合约模板市场, 查看模板详情, 根据模板一键创建合约工程



可视化调试





测试用例自动生成, 可视化增删用例, 一键调试并展示交易结果



网络配置

可视化配置区块链网络, 支持节点直连与通过Baas Rest连接

目前蚂蚁链智能合约开发平台支持多种技术栈进行开发，根据蚂蚁链内部团队的最佳实践，我们推荐以下几种开发技术栈组合：

- **后端 + 合约应用** (使用  ChainStack) : 后端应用 -> Java , 智能合约 -> C++
- **前端 + 合约应用** (使用  Myfish) : 前端应用 -> TypeScript, 智能合约 -> TypeScript
- **纯合约应用** (使用  CloudIDE /  应用速搭) : Solidity / C++

1. 环境配置

在 Cloud IDE 中，单击 **环境配置**，选择目标的合约链环境。



The screenshot displays the Cloud IDE interface for Ant Financial. At the top, it shows the Ant Financial logo and navigation options like '管理控制台' and '产品与服务'. The main workspace is divided into several sections:

- Compiler Settings:** Shows 'Solidity' as the language and '0.4.23' as the version, with a '编译' (Compile) button.
- Environment Configuration:** A button labeled '环境配置' (Environment Configuration) is highlighted with a red box and an arrow pointing to it.
- Contract Details:** Under the 'Voting' contract, it shows the bytecode, ABI, and deployment status. The ABI section includes a JSON object:


```
[{"constant":true,"inputs":[],"name":"totalVotes","outputs":[{"name":"","type":"uint256"}],"payable":false,
```
- Code Editor:** The right side shows the Solidity source code for 'demo.sol', including pragma statements and contract definitions for 'Voting'.
- Bottom Panel:** Contains a '编译' (Compile) button, tabs for '编译详情' (Compile Details), '合约分析' (Contract Analysis), and '调试详情' (Debug Details), and a 'Warning 3' indicator.

1.环境配置

在 **环境配置** 中，您可以选择已有的公共 **合约体验链**，该体验链会配备一个默认测试账户，可以用来部署、调用合约。

如果此测试账户不能满足需求，您也可以使用通过 SDK 创建的账户，但是需要填入私钥（16 进制字符串），因为在部署、调用合约时 IDE 都需要账户的私钥对交易做签名处理。

环境配置

选择链 

账户名

私钥

2. 编译合约

开发合约过程中，可以随时对已完成的代码进行编译。
在 Cloud IDE 中，单击 **编译**，开始编译代码。



The screenshot shows the Ant Financial Cloud IDE interface. The top navigation bar includes the Ant Financial logo, '金融科技' (Financial Technology), '管理控制台' (Management Control Panel), and '产品与服务' (Products and Services). The main editor area displays a Solidity contract named 'demo.sol' with the following code:

```
1 pragma solidity ^0.4.20;
2 // We have to specify which version of compiler this code will compile with,
3 // version should be lower than (or equals to) the supported version showing on this tool.
4 contract Voting {
5     /* Solidity doesn't let you pass in an array of strings in the constructor (yet).
6     We will use an array of bytes32 instead to store the list of candidates with string type.
7     */
8     bytes32[] public candidateList;
9
10    /* mapping field below is equivalent to an associative array or hash.
11    The key of the mapping is candidate name stored as type bytes32 and value is
12    an unsigned integer to store the vote count.
13    */
14    mapping (bytes32 => uint8) public votesReceived;
15
16    /* Counting for candidate counts and total votes for all candidates
17    */
18    uint256 public candidateCount;
19    uint256 public totalVotes;
20}
```

The bottom panel shows the compilation results, including a 'Warning 7' and several other warnings related to the compiler version and storage pointers.

3.部署合约

在合约编译通过，得到字节码和 ABI 后，在 Cloud IDE 中单击 **部署合约** 将合约部署到目标体验链进行测试。

在部署定义了 constructor 方法的合约时，需要给定 constructor 方法的参数。Cloud IDE 支持合约方法参数类型的提示和一些基本数据类型的初始化赋值，您可根据自身实际情况初始化合约。

部署合约

合约名称

合约参数

candidateNames - 0 + -

candidateNames - 1 + -

Gas预估

Gas限额

3.部署合约

部署成功后，可以看到合约的所有方法列表。



蚂蚁金服 | 金融科技 | 管理控制台 | 产品与服务

Solidity 0.4.23 编译 demo.sol

环境配置

05281019080805182019291905050806000

合约接口说明 (ABI) 已部署合约

```
[{"constant":true,"inputs": [],"name":"totalVotes","outputs":[{"name":"","type":"uint256"}],"payable":false,"stateMutability":"view","type":"function"}]
```

Voting0001

合约ID:0x1915501cf1587518d7a63d1ff541aa1...

TX Hash:0xf416a44c83b23537db582bb9fbf4e...

function totalVotes	调用合约
function totalVotesFor	调用合约
function validCandidate	调用合约
function votesReceived	调用合约
function candidateCount	调用合约
function candidateList	调用合约
function voteForCandidate	调用合约

```
1 pragma solidity ^0.4.20;
2 // We have to specify which version of compiler this code will compile with,
3 // version should be lower than (or equals to) the supported version showing on this tool.
4 contract Voting {
5     /* Solidity doesn't let you pass in an array of strings in the constructor (yet).
6     We will use an array of bytes32 instead to store the list of candidates with string type.
7     */
8     bytes32[] public candidateList;
9
10    /* mapping field below is equivalent to an associative array or hash.
11    The key of the mapping is candidate name stored as type bytes32 and value is
12    an unsigned integer to store the vote count.
13    */
14    mapping (bytes32 => uint8) public votesReceived;
15
16    /* Counting for candidate counts and total votes for all candidates
17    */
18    uint256 public candidateCount;
19    uint256 public totalVotes;
20}
```

编译

编译详情 合约分析 调试详情

Warning 7

- Warning: This is a pre-release compiler version, please do not use it in production.
- demo.sol:42:9: Warning: Variable is declared as a storage pointer. Use an explicit "storage" keyword. `uint [] array;`
- demo.sol:42:9: Warning: Uninitialized storage pointer. Did you mean '<type> memory array'?
- demo.sol:43:9: Warning: Function declared as view, but this expression (potentially) modifies the state of the contract. `array.push(1);`
- demo.sol:44:9: Warning: Function declared as view, but this expression (potentially) modifies the state of the contract. `array.push(2);`

保存

4.升级合约 (可选)

前置条件

- 请确保您要升级的合约代码已在Cloud IDE中执行过编译并编译成功。
- 合约升级仅支持正式发布的合约，请确保在部署合约时已打开部署合约对话框中的是否正式发布的开关。
- 请确保升级合约的账户与部署合约时的账户一致。

部署合约

是否正式发布

合约名称

合约参数

candidateNames +

Gas预估

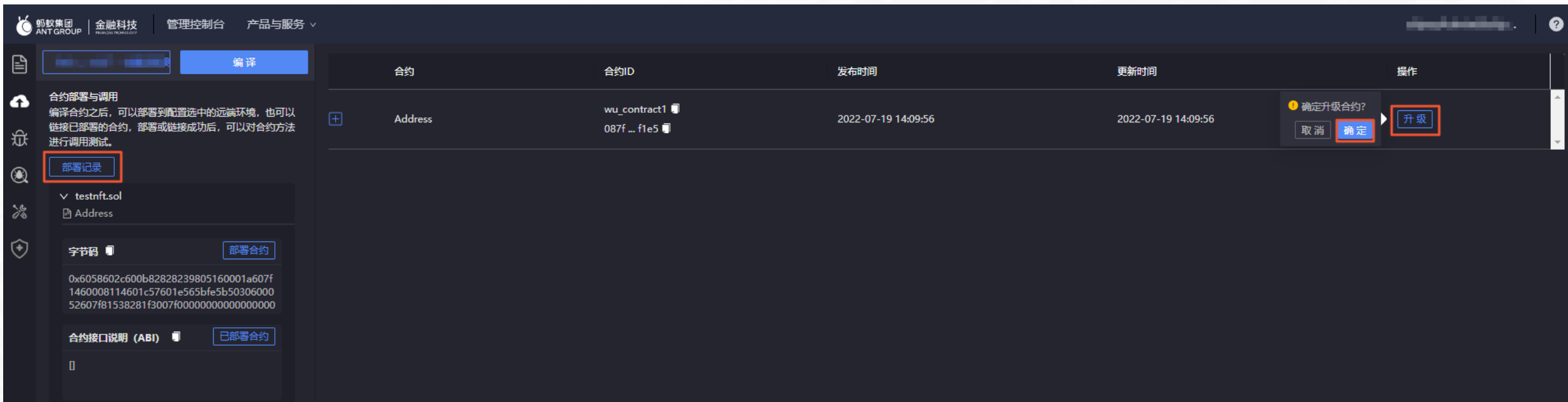
Gas限额

确定

4.升级合约 (可选)

对于正式发布后的合约，如果您需要在Cloud IDE中进行合约升级，可依照如下步骤进行操作：

- 1.点击合约下方的**部署记录**按钮，查看所有正式发布的合约。
- 2.在正式发布的合约中找到您需要升级的合约，并点击记录右侧的**升级**按钮。



The screenshot displays the Cloud IDE interface for contract management. On the left sidebar, the 'Deployment Records' (部署记录) button is highlighted with a red box. The main area shows a table of contracts with columns for 'Contract' (合约), 'Contract ID' (合约ID), 'Release Time' (发布时间), 'Update Time' (更新时间), and 'Action' (操作). A contract named 'Address' is selected, and a modal dialog is open with the question 'Confirm upgrade contract?' (确定升级合约?). The 'Upgrade' (升级) button in the dialog is highlighted with a red box.

合约	合约ID	发布时间	更新时间	操作
Address	wu_contract1 087f ... f1e5	2022-07-19 14:09:56	2022-07-19 14:09:56	确定升级合约? 取消 确定 升级

4.升级合约 (可选)

3.在弹出的合约升级对话框中，单击确定，执行合约升级。

合约升级

Gas预估

Gas限额

4.合约升级完成后会弹出通知提示，表示新的合约字节码已经完成上链。

合约	合约ID	发布时间	更新时间	操作
<input type="button" value="+"/> Address	wu_contract1  087f ... f1e5 	2022-07-19 14:09:56	2022-07-19 14:13:32	<input type="button" value="升级"/>

5.调用合约

合约部署成功后，IDE 会列出合约中所有的 public 方法以及 public 类型的状态变量。单击目标方法右侧的 调用合约 进行调用。



The screenshot shows the Ant Financial IDE interface. On the left, the 'Voting0001' contract is displayed with its ID and TX Hash. Below this, a list of public functions is shown, each with a '调用合约' (Call Contract) button. The 'candidateCount' function is highlighted with a red box, and its call parameters are shown: tx hash (0xeb985c1576adf731d1a6610030be5c9...), input (empty), output (uint256: 2), and log (empty). Other functions listed include totalVotes, totalVotesFor, validCandidate, votesReceived, candidateList, and voteForCandidate. On the right, the Solidity source code for 'demo.sol' is shown, including pragma, contract definition, and state variables. At the bottom right, a 'Warning 7' section lists several compiler warnings related to pre-release compiler versions and uninitialized storage pointers.

错误说明

在使用 Cloud IDE 过程中，可能遇到一些错误提示。在调用合约时，如果合约执行过程中出错，可能有多种原因，Cloud IDE 会将合约调用失败的错误码信息返回，便于您分析原因。Cloud IDE 使用的错误码信息与 SDK 使用的错误码保持一致，可以查看错误码列表找到对应的错误码说明。



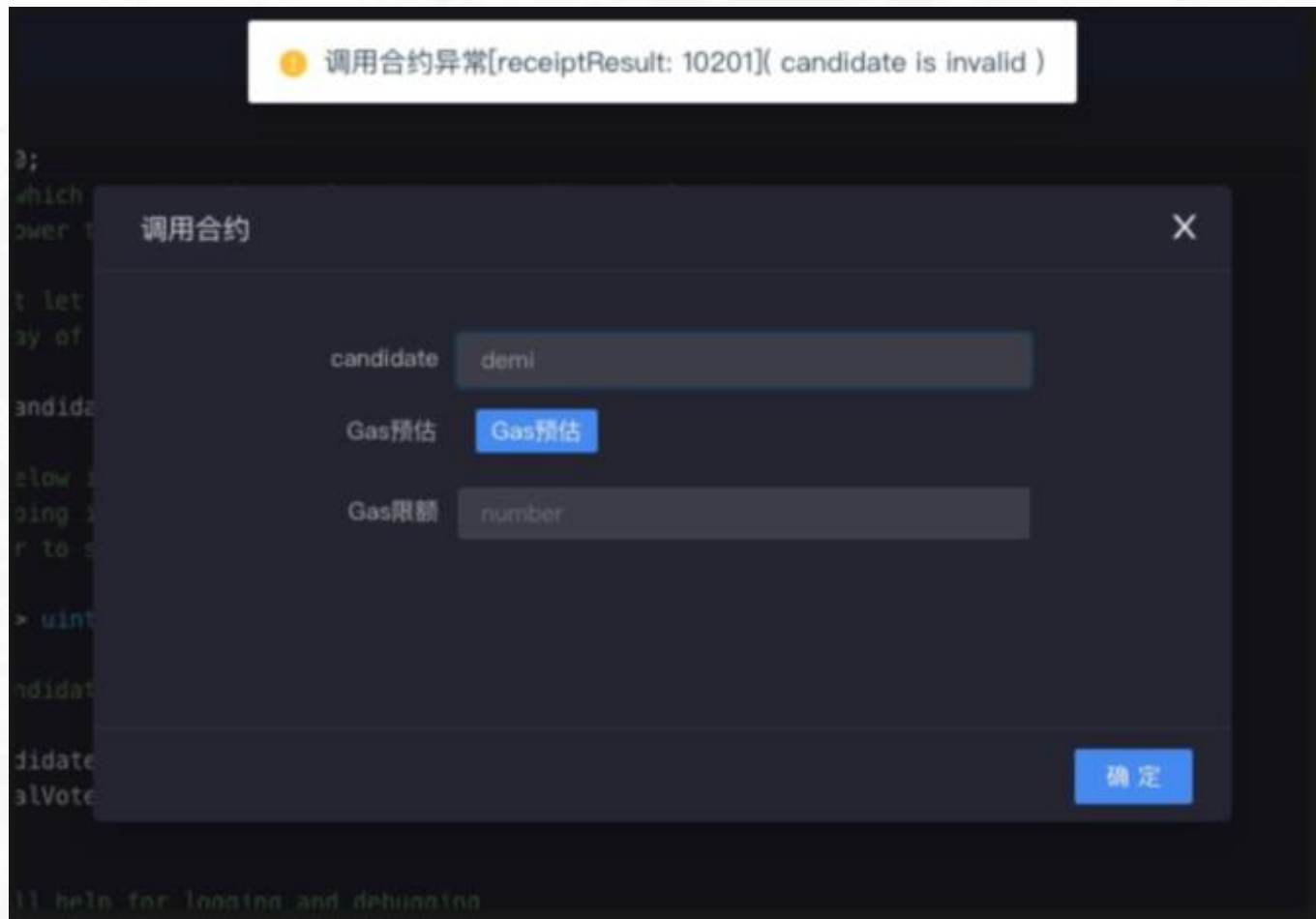
在右图的错误码示例中，系统调用了 Cloud IDE 示例合约的 `voteForCandidate` 方法，给定的候选人为 `Demi`，返回的错误码为 `10201`，其含义为：

错误码	错误码值	描述
VM_REVERT	10201	触发 revert 指令导致的错误。

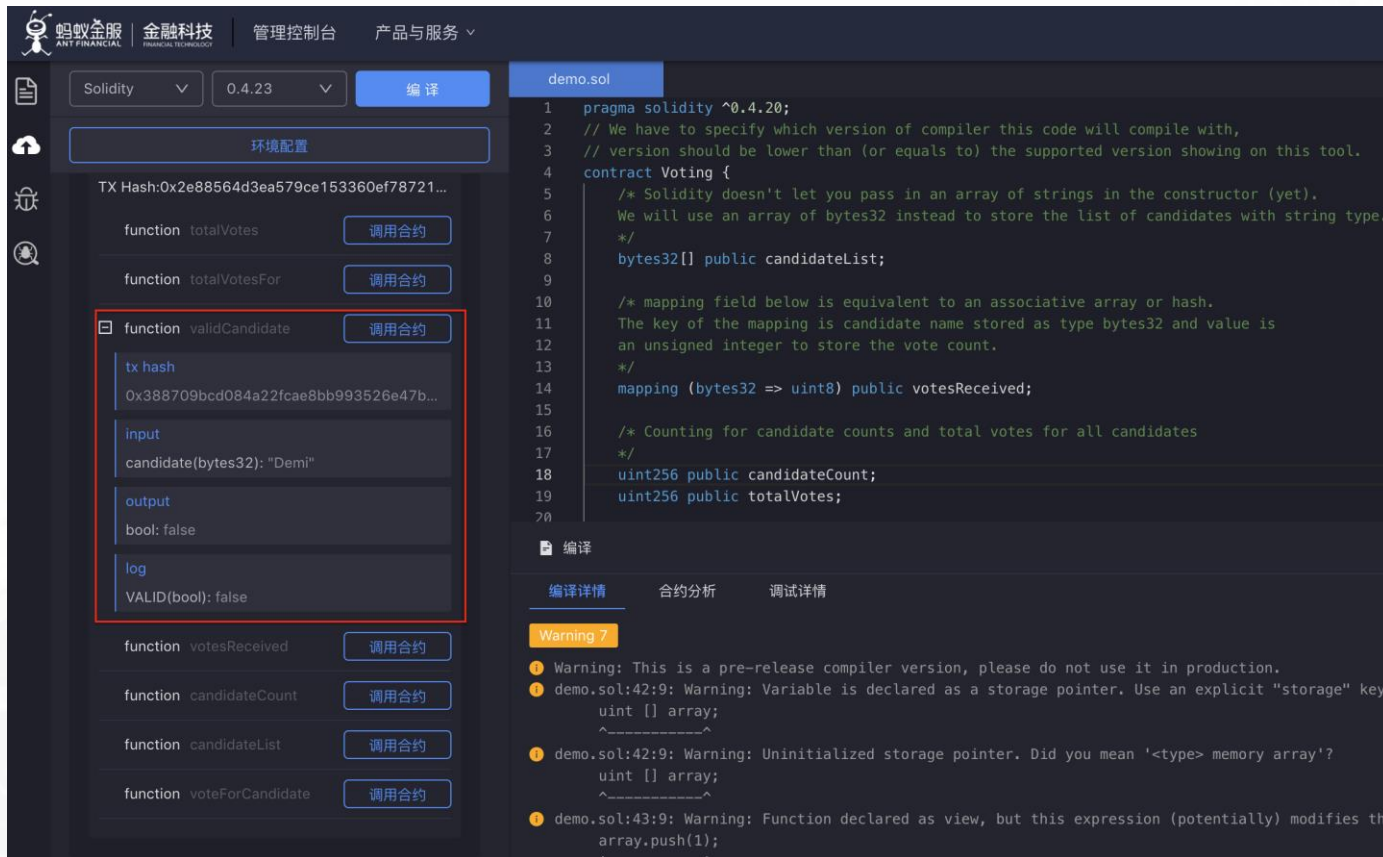
根据错误码提示，可以判断调用合约失败是因为触发了 `revert` 语句，含义为某个 `require()` 方法判断失败。查看合约代码，即可发现 `voteForCandidate` 方法中的相关条件检查：

```
require(validCandidate(candidate));
```

由此，可推断出输入的候选人 `Demi` 不合法，并不在实际投票的候选人列表 `candidateList` 中。事实也确实如此，合约部署过程中，`constructor` 参数仅指定了 `Nick` 和 `Rose` 两个候选人参与竞选。



可以通过添加 Event 事件触发日志的方式实现逻辑分支的判断和分析，进而对合约内部逻辑进行调试。
这里选择调用合约方法 validCandidate。



```
1 pragma solidity ^0.4.20;
2 // We have to specify which version of compiler this code will compile with,
3 // version should be lower than (or equals to) the supported version showing on this tool.
4 contract Voting {
5     /* Solidity doesn't let you pass in an array of strings in the constructor (yet).
6     We will use an array of bytes32 instead to store the list of candidates with string type.
7     */
8     bytes32[] public candidateList;
9
10    /* mapping field below is equivalent to an associative array or hash.
11    The key of the mapping is candidate name stored as type bytes32 and value is
12    an unsigned integer to store the vote count.
13    */
14    mapping (bytes32 => uint8) public votesReceived;
15
16    /* Counting for candidate counts and total votes for all candidates
17    */
18    uint256 public candidateCount;
19    uint256 public totalVotes;
20}
```

// This function will help to check whether target candidate is in the candidateList.

```
function validCandidate(bytes32 candidate) view public returns (bool) {
    for (uint i = 0; i < candidateList.length; i++) {
        if (candidateList[i] == candidate) {
            emit VALID(true);
            return true;
        }
    }
    emit VALID(false);
    return false;
}
```

在 validCandidate 方法的实现中，不同的执行逻辑会根据情况触发不同的 Event 事件，通过查看调用结果日志信息，即可分析出合约的执行路径。

在以上示例中，执行逻辑相对简单。在复杂的逻辑控制方法中，通过添加 Event 事件的方式可以很好的追踪合约执行路径，达到调试合约的目的。

合约单步调试

合约调试模块增加了一个虚拟机模拟器，可以模拟运行 Solidity 合约并给出执行结果。当前虚拟机模拟器与真实合约链有一些差异，主要差异在于虚拟机模拟器暂时不支持对 JSON/XML 的解析，将在后续功能升级中提供支持。



通常，对一个交易的 Debug 过程如下：

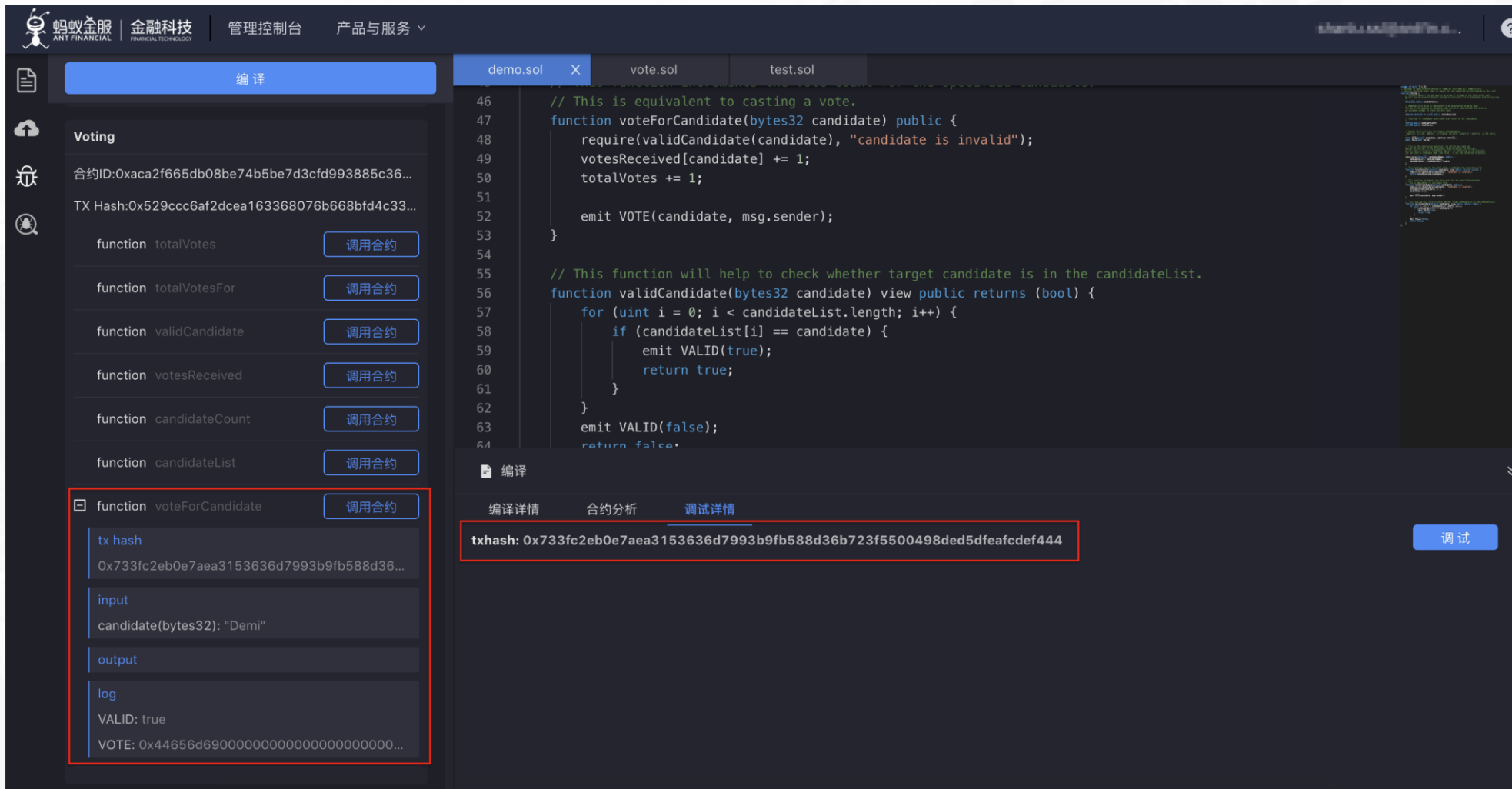
1. 编译并部署合约到虚拟机模拟器中。

下图示例中，部署创建工程时默认添加的 demo 样例合约，部署时初始化了两个候选人 “Demi” 和 “Rose”：

The screenshot shows a dark-themed dialog box titled "部署合约" (Deploy Contract). It contains the following fields and controls:

- 合约名称 (Contract Name): Voting
- 合约参数 (Contract Parameters):
 - candidateNames - 0: Demi
 - candidateNames - 1: Rose
- Buttons: A blue "确定" (Confirm) button at the bottom right, and plus/minus icons next to each parameter field.

2.调用要调试的目标合约方法，产生执行的交易 hash。
下图示例中，调用默认 demo 样例的方法 voteForCandidate:



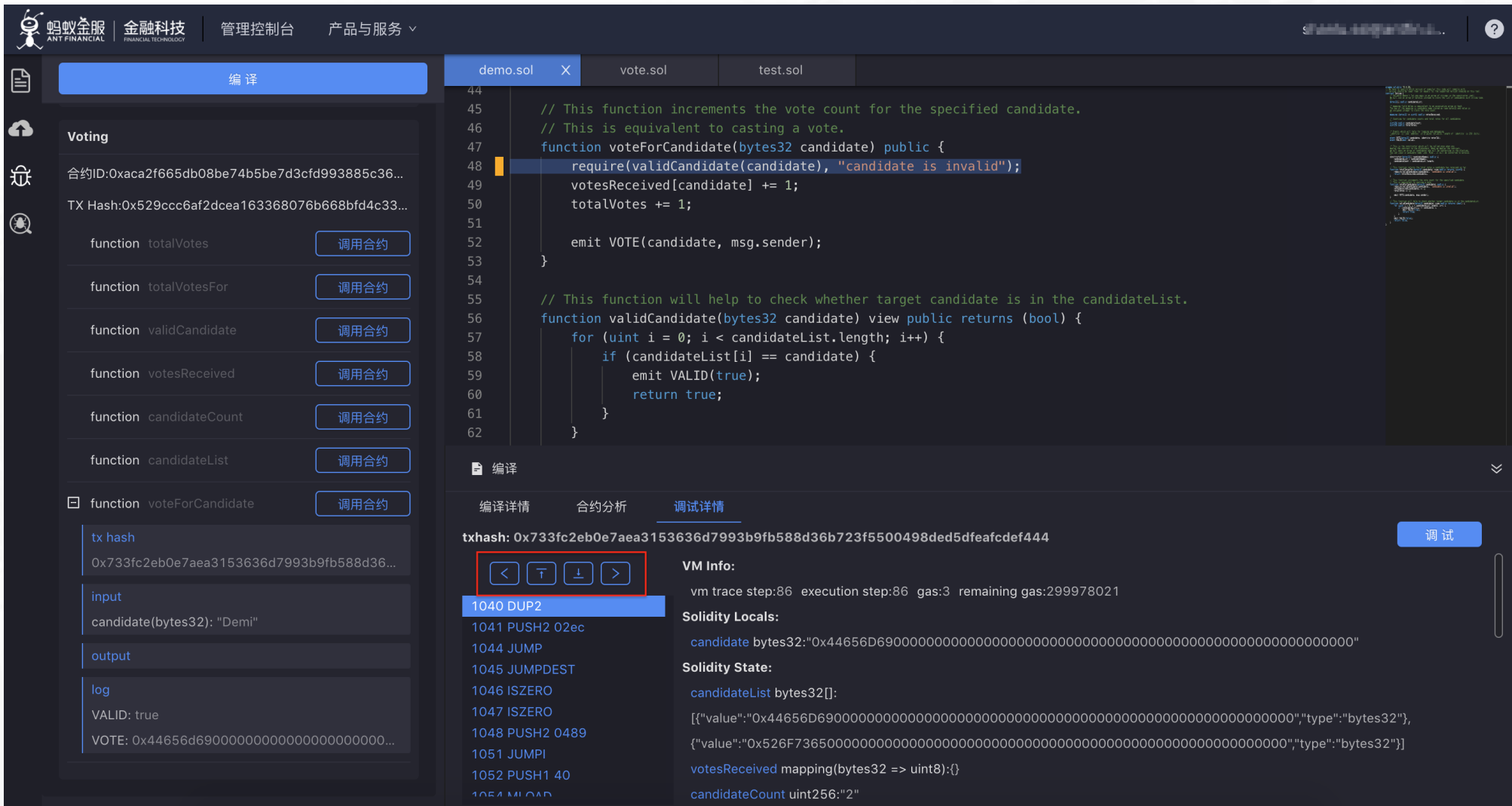
The screenshot displays the Ant Financial smart contract debugging interface. On the left, a list of functions is shown, with 'function voteForCandidate' selected. The main area displays the Solidity code for 'voteForCandidate' and 'validCandidate'. The bottom panel shows the transaction hash (txhash) and a '调试' (Debug) button.

```
46 // This is equivalent to casting a vote.
47 function voteForCandidate(bytes32 candidate) public {
48     require(validCandidate(candidate), "candidate is invalid");
49     votesReceived[candidate] += 1;
50     totalVotes += 1;
51
52     emit VOTE(candidate, msg.sender);
53 }
54
55 // This function will help to check whether target candidate is in the candidateList.
56 function validCandidate(bytes32 candidate) view public returns (bool) {
57     for (uint i = 0; i < candidateList.length; i++) {
58         if (candidateList[i] == candidate) {
59             emit VALID(true);
60             return true;
61         }
62     }
63     emit VALID(false);
64     return false;
65 }
```

txhash: 0x733fc2eb0e7aea3153636d7993b9fb588d36b723f5500498ded5dfeafcdf444

调试

- 3.在底部 调试详情 信息栏，可单击 调试 按钮直接调试步骤 2 产生的目标交易。
- 4.可以单步单击图示中的箭头按钮，进行单步调试，同时查看运行时的局部变量、合约存储状态、堆栈、内存等关键信息进行深入调试。



蚂蚁金服 金融科技 管理控制台 产品与服务

demo.sol x vote.sol test.sol

```
44
45 // This function increments the vote count for the specified candidate.
46 // This is equivalent to casting a vote.
47 function voteForCandidate(bytes32 candidate) public {
48     require(validCandidate(candidate), "candidate is invalid");
49     votesReceived[candidate] += 1;
50     totalVotes += 1;
51
52     emit VOTE(candidate, msg.sender);
53 }
54
55 // This function will help to check whether target candidate is in the candidateList.
56 function validCandidate(bytes32 candidate) view public returns (bool) {
57     for (uint i = 0; i < candidateList.length; i++) {
58         if (candidateList[i] == candidate) {
59             emit VALID(true);
60             return true;
61         }
62     }
63 }
```

编译

合约ID:0xaca2f665db08be74b5be7d3cfd993885c36...
TX Hash:0x529ccc6af2dcea163368076b668bfd4c33...

function totalVotes 调用合约
function totalVotesFor 调用合约
function validCandidate 调用合约
function votesReceived 调用合约
function candidateCount 调用合约
function candidateList 调用合约
function voteForCandidate 调用合约

tx hash
0x733fc2eb0e7aea3153636d7993b9fb588d36...

input
candidate(bytes32): "Demi"

output

log
VALID: true
VOTE: 0x44656d69000000000000000000000000...

编译详情 合约分析 调试详情

txhash: 0x733fc2eb0e7aea3153636d7993b9fb588d36b723f5500498ded5dfeafcd444

调试

VM Info:
vm trace step:86 execution step:86 gas:3 remaining gas:299978021

Solidity Locals:
candidate bytes32:"0x44656d6900"

Solidity State:
candidateList bytes32[]:
[{"value":"0x44656d6900","type":"bytes32"},
{"value":"0x526f736500","type":"bytes32"}]
votesReceived mapping(bytes32 => uint8):{}
candidateCount uint256:"2"

1040 DUP2
1041 PUSH2 02ec
1044 JUMP
1045 JUMPDEST
1046 ISZERO
1047 ISZERO
1048 PUSH2 0489
1051 JUMPI
1052 PUSH1 40
1054 MLOAD

本章小结

本章主要介绍蚂蚁链技术平台及应用。

- 第1节介绍了区块链技术的发展现状及展望，并概述了蚂蚁链技术平台及业务。
- 第2节介绍了蚂蚁链的技术优势和亮点示例。
- 第3节展示了一些蚂蚁链与传统行业结合的产业实践。
- 第4节介绍了蚂蚁链开发者实验室，包括DApp开发应用框架、DApp开发工具，并以Cloud IDE为例展示了开发智能合约的过程。



北京大学
PEKING UNIVERSITY

感谢观看

