

# 区块链构建时序数据可信上链

华为云 张子怡



# 目录

CONTENTS



01

趋势挑战

02

架构方案

03

开源介绍



# 区块链的发展的3个阶段



## 公有链为主

最初起源于比特币，提供真实安全的数字货币交易功能

## 联盟链为主

通过智能合约，在政务、金融、制造、医疗等多个应用场景下实现价值的可信传递

## 数字经济基础设施

区块链与云计算、大数据和人工智能等新兴技术交叉演进，将重构数字经济发展生态，促进价值互联网与实体经济的深度融合。

区块链1.0  
虚拟货币



区块链2.0  
企业应用



区块链3.0  
价值互联网



我们在这

时间轴

# 区块链价值：促共享，优流程，降成本，提效率，建设可信体系



要抓住区块链技术融合、功能拓展、产业细分的契机，发挥区块链在促进数据共享、优化业务流程、降低运营成本、提升协同效率、建设可信体系等方面的作用。

-----习近平

区块链应用场景



生态构建



共建以城市为单位的可信互通体系

数据开放

数据协同

数据共享

数据归集

应用生态

开放生态

协同生态

共享生态

汇聚生态

政务

金融

工业

生活

可信协同平台

多链协同平台

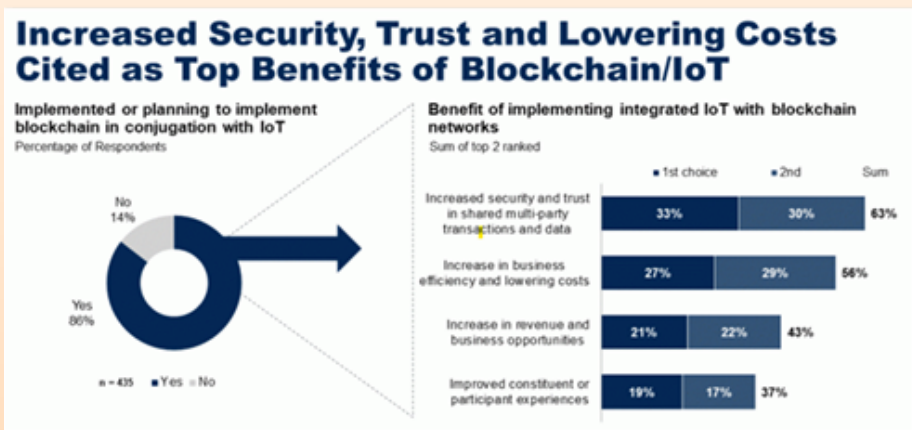
万物链接

# 区块链在IOT领域的需求和前景



## IoT integration is a Sweet Spot for Blockchain Per Gartner Survey

75% of IoT technology adopters in the U.S. have already adopted blockchain or are planning to adopt it by the end of 2020. Among the blockchain adopters, 86% are implementing the two technologies together in various projects.

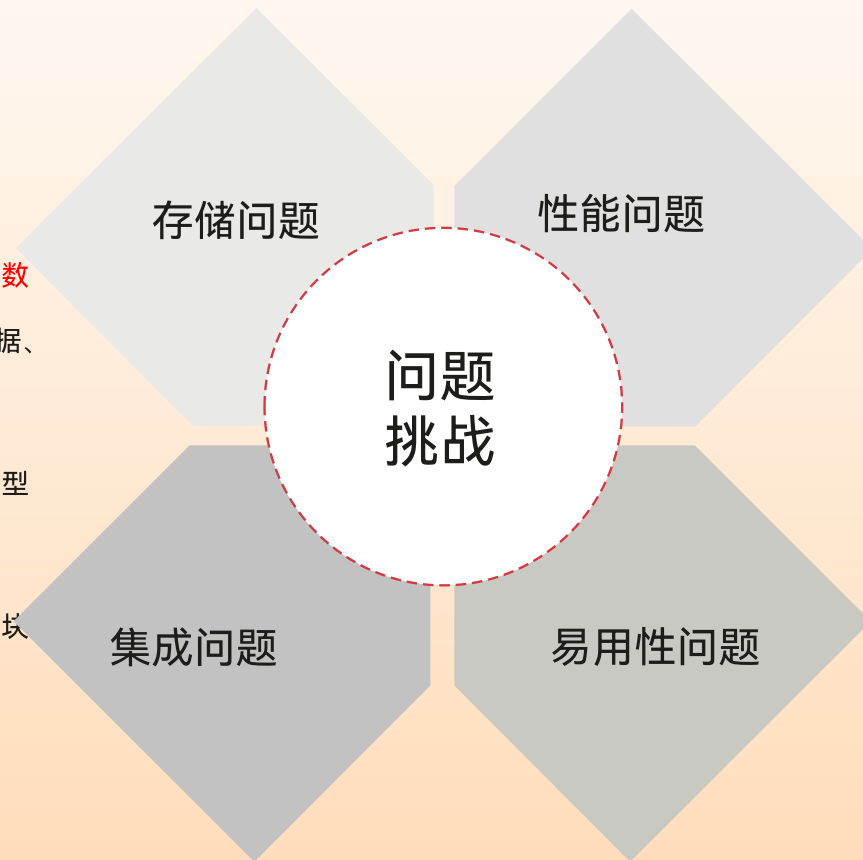


- 2021年全球物联网（企业级）支出规模达6902.6亿美元，并有望在2026年达到1.1万亿美元 —— IDC, 《全球物联网支出指南2022V1》
- 友商情况
  - 国内
    - 工业品溯源解决方案
    - 物联网设备可信上链方案（IoT模组）
  - 全球
    - IOTA: 面向IoT场景的区块链平台，基于DAG共识提供高性能交易能力
    - Filament: 面向IoT设备的区块链套件（身份安全、上链等）

# 当前区块链结合IOT技术普遍面临的问题和挑战



- **性能问题**: 交易性能低、规模扩展难
- **易用性问题**: 仅支持KV类型的数据存储和读取接口, 或针对预先指定类型数据的增强查询能力, 无法支持针对其他复杂数据类型 (关系型数据、流式数据、时序数据等) 的OLTP (基础操作)、OLAP (进阶分析) 能力
- **存储问题**: 仅支持KV类型数据存储, 无法支持复杂类型数据的存储 (关系型数据、流式数据、时序数据等)
- **集成问题**: 如需要支持复杂类型的数据处理和分析能力, 则需侵入式修改区块链平台的代码, 这是一项非常艰难的开发工作, 且面临开源代码回合的问题



高效、安全、易用、开放

# 目录

CONTENTS



01

趋势挑战

02

架构方案

03

开源介绍



# IOT区块链时序服务 (tempo) 架构

面向现有区块链平台的能力增强工具（插件） - 针对时序数据场景

通过一种**非侵入式**的方法，在不破坏区块链**去中心化**特性的前提下，拓展现有区块链平台面向时序数据的存储和处理能力，使得区块链能够支撑更多样业务场景，主要是IoT场景

区块链时序数据服务



## 技术点

- 1 针对不同类型区块链平台提供不同类型的连接器，如Hyperledger Fabric平台使用外置链码型，以太坊平台使用预言机型。设计上均需保证去中心化
- 2 针对不同的时序数据库提供不同的连接器。由于各时序数据库仅为增强工具提供存储和分析能力支撑，因此
- 3 通过引入下层时序数据库，上层智能合约可植入面向时序数据的方法，包括时序数据存储、数据汇聚、分析算子执行等
- 4 提供区块链网关将数据上链。区块链网关可以从IoT网关消费数据，也可与IoT设备直连，针对海量数据，采用Rollup机制对数据进行打包
- 5 由于智能合约无定时触发能力，提供外部合约触发器，为用户提供定时报表生成能力
- 6 提供一套易用的DSL，通过定义数据特征与预期汇聚分析规则，可自动生成区块链网关、智能合约及合约触发器

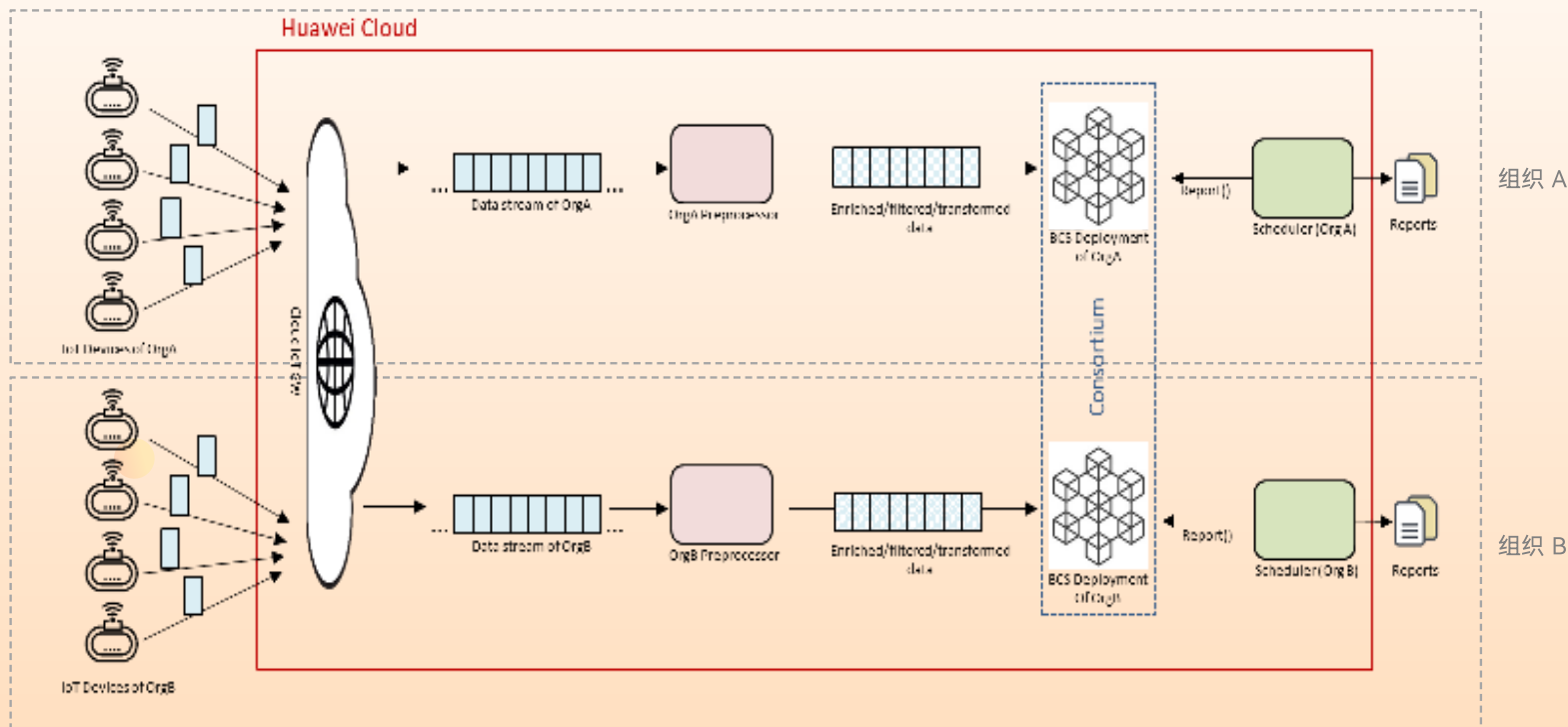


# 物联网数据预处理管道为海量设备接入提供基础

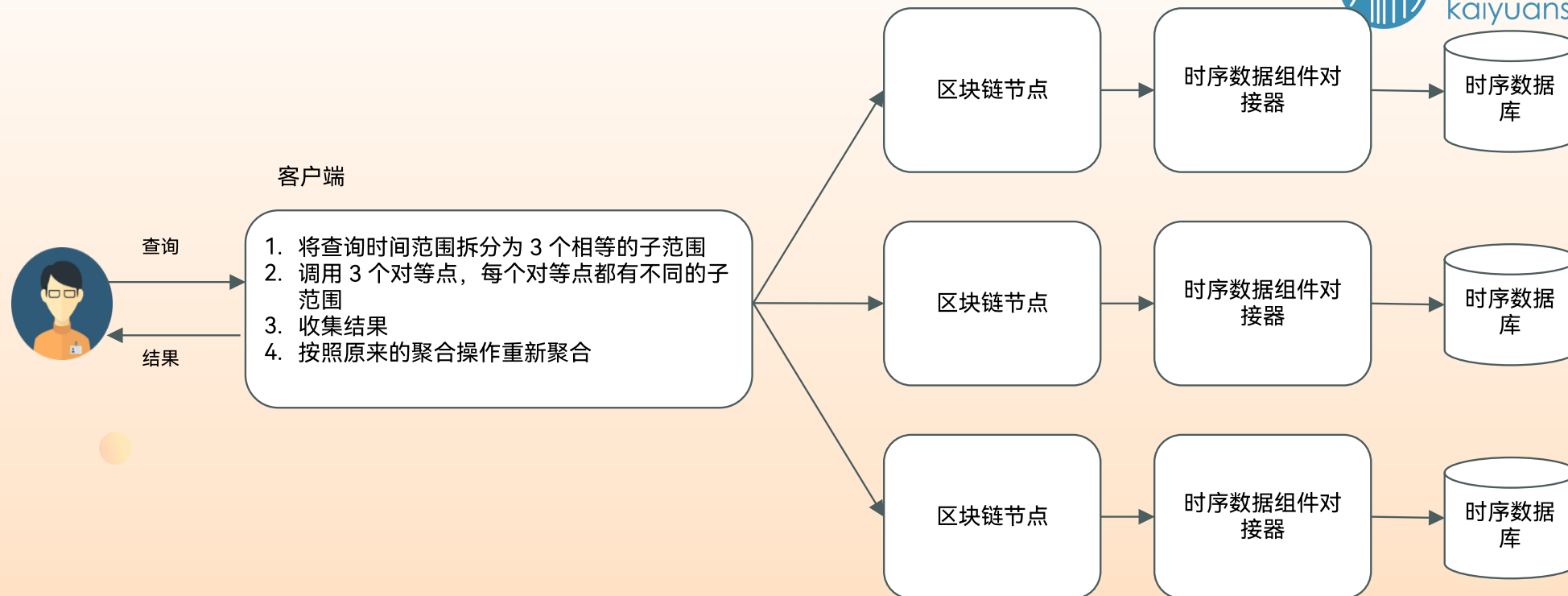


## 关键技术

- **数据对齐**——例如一些物联网设备以摄氏度为单位测量，一些以华氏度为单位。预处理将所有测量值调整为摄氏度。
- **数据减少**——物联网设备是一个监控摄像头，发送的视频文件太大而无法存储在账本中。预处理调用外部服务从镜头中提取特定数据，只有该数据才会提交到区块链。
- **数据聚合**——物联网设备每 10 毫秒报告一次数据。联盟成员只对最后一秒的测量感兴趣。预处理可以聚合上一秒的测量值并提交聚合。
- **数据批处理**——利用 TS DB 的高效批量插入功能



# IOT 数据并行高效查询能力



- 依赖于账本数据库在所有对等点之间完全复制
- 当查询聚合操作支持重新聚合 (例如总和、计数、乘积等)

# 并行高效执行调用客户端样例



```
ClientExampleSimple.java
21 public class ClientExampleSimple {
22
23     public static void main(String[] args) throws CertificateException, IOException, InvalidKeyException {
24         try (Gateway gateway = buildGateway().connect()) {
25             Contract contract = gateway.getNetwork("channel1").getContract("asset1", "com.saferoads.chaincode.SubscriberRegistration");
26             // Query Data
27             List<CompletableFuture<byte[]>> asyncList = new ArrayList<>();
28             int numberOfSplits = 4;
29             int maxTime = 100000000;
30
31             for (int i = 0; i < numberOfSplits; i++) {
32                 Map<String, Object> bindParameters = new HashMap<>();
33                 //create splits of time range
34                 int start = i * (maxTime / numberOfSplits) + 1;
35                 int end = (i + 1) * (maxTime / numberOfSplits);
36                 bindParameters.put("start", "time(v:" + start + ")");
37                 bindParameters.put("stop", "time(v:" + end + ")");
38                 bindParameters.put("plate", "1");
39                 ObjectWriter ow = new ObjectMapper().writer().withDefaultPrettyPrinter();
40                 String json = ow.writeValueAsString(bindParameters);
41
42                 System.out.println(json);
43
44                 //Invoke the splits in parallel
45                 CompletableFuture<byte[]> completableFuture = CompletableFuture.supplyAsync(() -> {
46                     try {
47                         return contract.evaluateTransaction("runTollRoadCameraReports", json);
48                     } catch (ContractException e) {
49                         throw new RuntimeException(e);
50                     }
51                 });
52                 asyncList.add(completableFuture);
53             }
54
55             // Re-aggregate the results
56             int count = 0;
57             for (int i = 0; i < numberOfSplits; i++) {
58                 int current = Integer.parseInt(new String(asyncList.get(i).get(), StandardCharsets.UTF_8));
59                 System.out.println(current);
60                 count += current;
61             }
62             System.out.println("Final:");
63             System.out.println(count);
64
65         } catch (ExecutionException | InterruptedException e) {
66             throw new RuntimeException(e);
67         }
68     }
69 }
```

如果您的组织没有足够的节点，您可以创建一个调用任何节点的自定义查询构建器，如下所示：

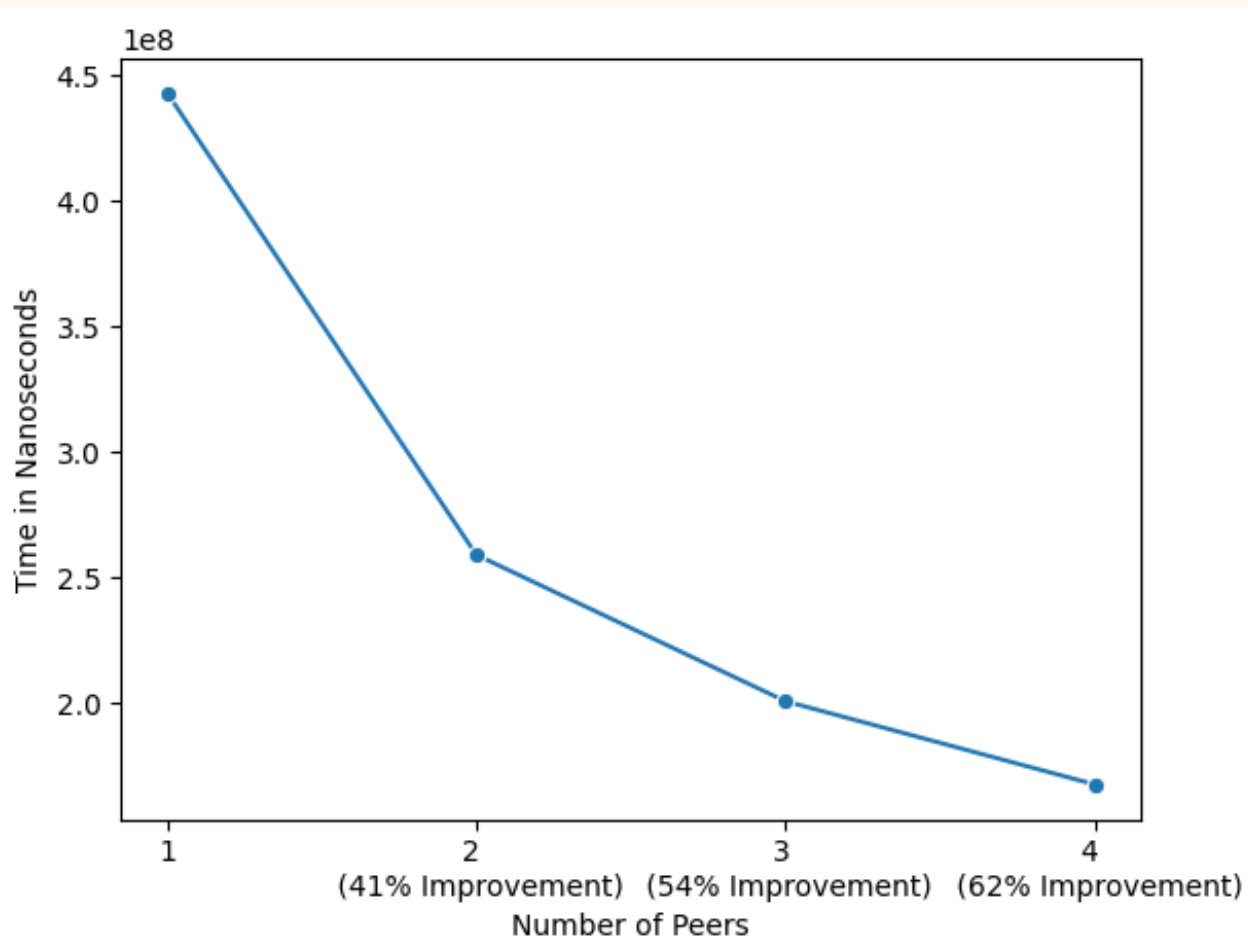
```
ClientExampleSimple.java | CustomQueryHandlerFactory.java
package com.huawei.chronix.example;

import org.hyperledger.fabric.gateway.Network;
import org.hyperledger.fabric.gateway.impl.query.RoundRobinQueryHandler;
import org.hyperledger.fabric.gateway.spi.QueryHandler;
import org.hyperledger.fabric.gateway.spi.QueryHandlerFactory;

public class CustomQueryHandlerFactory implements QueryHandlerFactory {
    @Override
    public QueryHandler create(Network network) {
        return new RoundRobinQueryHandler(network.getChannel().getPeers());
    }
}
```

# 并行查询执行基准测试结果

- 通过 2 个对等点并行查询，性能提高了 41%
- 将查询并行化 3 个对等结果提高了 54%
- 在 4 个对等点上并行查询结果提高了 62%



# 目录

CONTENTS



01

趋势挑战

02

架构方案

03

开源介绍



# 区块链领域开源生态介绍



## ■ 区块链的开放本质

- 区块链的本质是一个去中心化的分布式账本数据库，依托本身的技术机制确保在一个半信任的使用场景下，数据库中的内容不可篡改，且一个区块链网络中的各参与方数据一致。
- **开源可以保证区块链平台的技术透明、可信**，同时作为一个旨在达成去中心化的技术，本身的架构演进应当尽可能去中心化或多中心化，而开源社区就是实现这一需求天然的媒介。

## ■ 全球区块链平台开源情况

- 公链：从区块链走向大众的奠基者比特币，到后续的集大成者以太坊、EOS等，均将源码贡献给开源社区，由社区开发者共同维护以完善各自能力，同时通过开源构建各自平台的可信度；
- 联盟链（企业区块链）：
  - 国外：诸如Hyperledger Fabric、Quorum、Corda等头部平台均已开放源代码；
  - 国内：对待开源的态度不一，诸如FISCO BCOS、XuperChain\ HyperChain等平台已开放源代码。

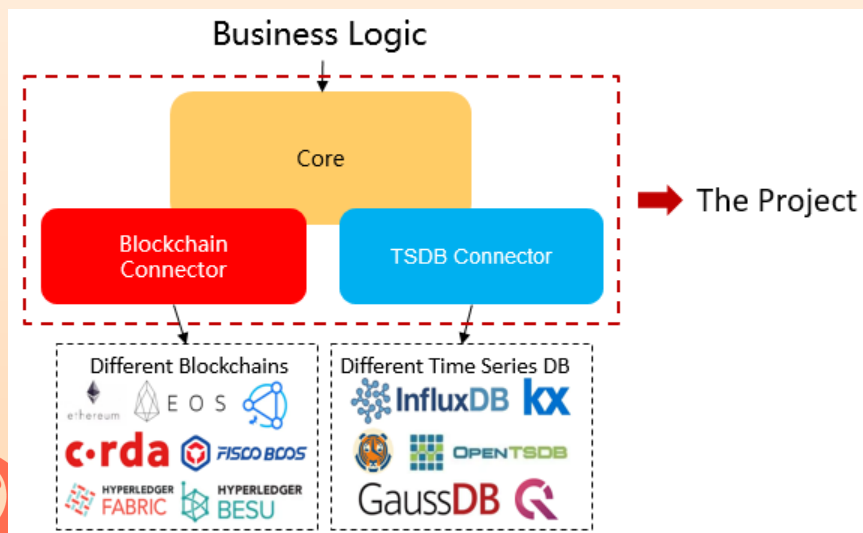


# Tempo技术架构开源方案



- IoT + 区块链场景的需求
  - 支持存储Timepoint类型的数据，并可通过智能合约进行数据写入、数据读取、数据汇聚和数据分析
  - 针对时间线长、数据量大的数据汇聚和分析任务，可依赖区块链多点存储的特性，通过并行计算加速任务的执行，支撑时延敏感的业务场景（交通治理等）

通过一种**非侵入式**的方法，在**不破坏区块链去中心化特性**的前提下，**拓展**现有区块链平台面向时序数据的**存储和处理能力**，使得区块链能够支撑更多**业务场景**，包括**IoT场景**



## 核心技术点

新增若干组件，具备：

1. 逻辑执行能力（类智能合约）
2. 插件化连接不同区块链平台的能力
3. 插件化连接不同数据库的能力

■ Huawei Component: 核心模块，负责接收用户请求，并具备预设逻辑执行能力、调动两个连接器的能力

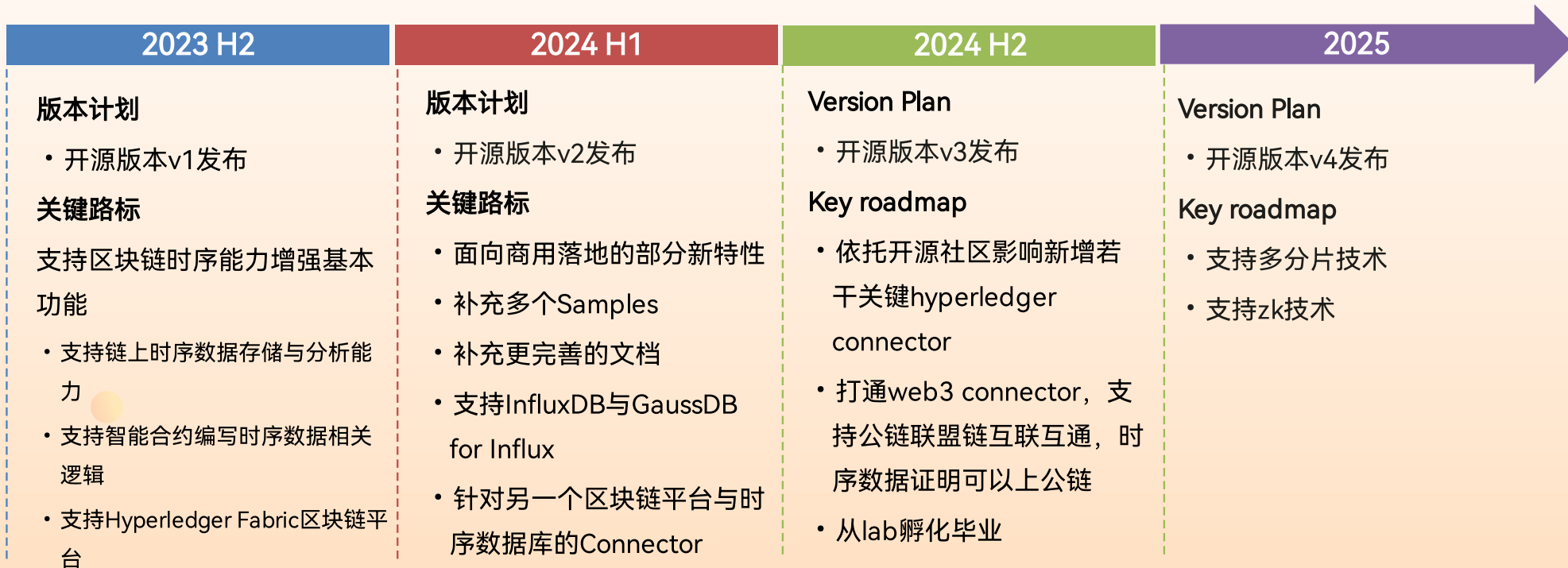
■ Blockchain Connector: 连接不同类型的区块链平台，包括主要面向联盟链场景的类Hyperledger Fabric平台与主要面向公链场景的类以太坊平台

■ TSDB Connector: 连接不同类型的时序数据库，包括InfluxDB、Cassandra等

# 开源计划&路标



- 正在Hyperledger社区立项，进入LAB





# THANK YOU

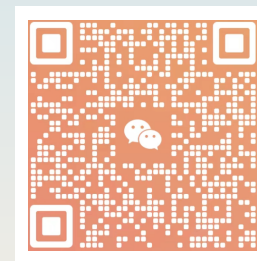
## QUESTIONS?



欢迎扫码打卡  
积分可兑换对应礼品哟!



扫码关注开源社公众号



扫码添加讲师联系方式

微信公众号：开源社KAIYUANSHE

视频号：开源社KAIYUANSHE

新浪微博：开源社

B站：开源社KAIYUANSHE

简书：开源社

头条：开源社

Facebook: KaiyuansheChina

Twitter: 开源社KAIYUANSHE