

金融业务系统架构设计与开发最佳实践

证券 · 外汇 · 交易银行 · 债券 · 工商银行业务深度解析

Domain Securities

Domain FX Trading

Domain Transaction Banking

Domain Bonds

Domain Corporate Banking

全面解析五大专业金融业务领域系统架构、技术选型与开发最佳实践

目录

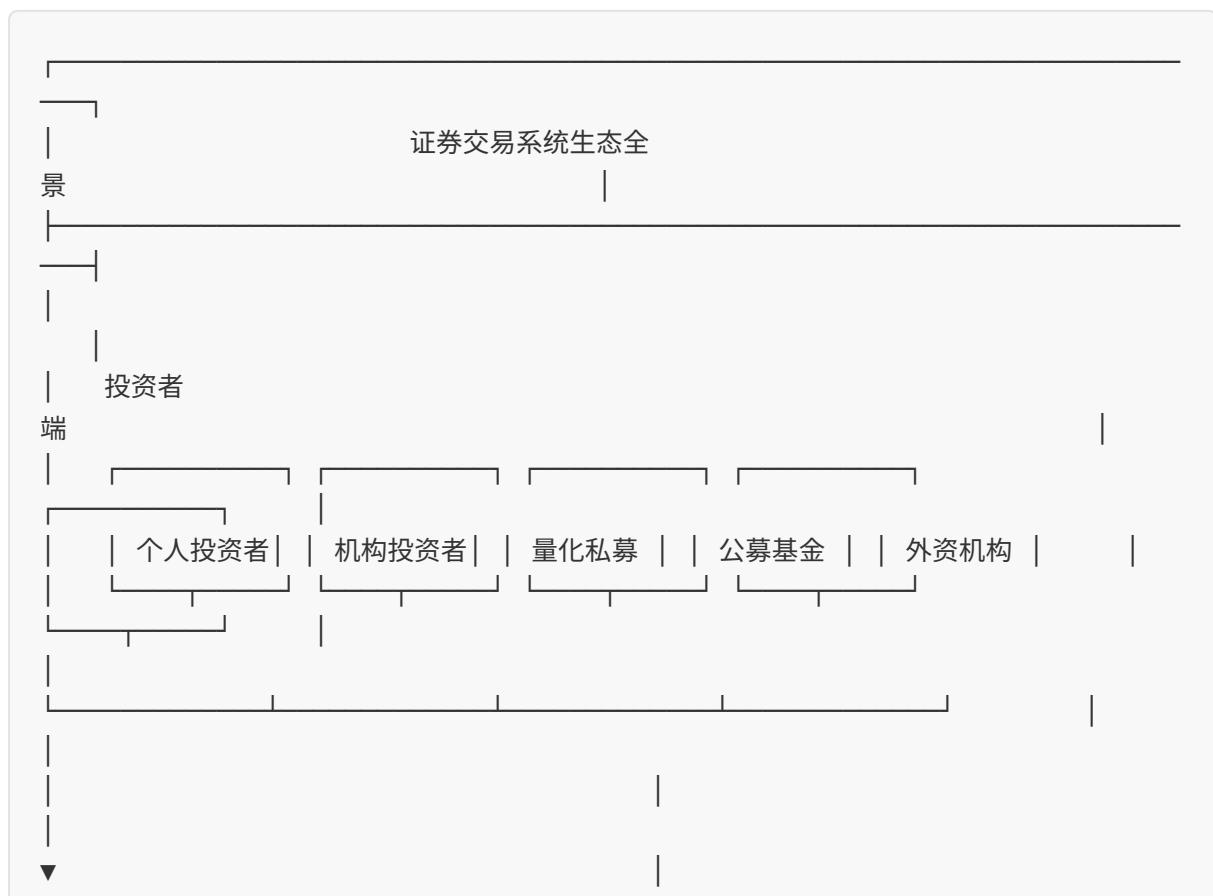
- [第一部分：证券业务系统](#)
- [第1章：证券核心交易系统](#)
- [第2章：证券清算结算系统](#)
- [第3章：证券风控与合规](#)
- [第二部分：外汇业务系统](#)
- [第4章：外汇交易系统](#)
- [第5章：跨境支付与清算](#)
- [第6章：外汇风险管理](#)
- [第三部分：交易银行业务系统](#)
- [第7章：现金管理系统](#)
- [第8章：供应链金融系统](#)
- [第9章：跨境金融平台](#)
- [第四部分：债券业务系统](#)

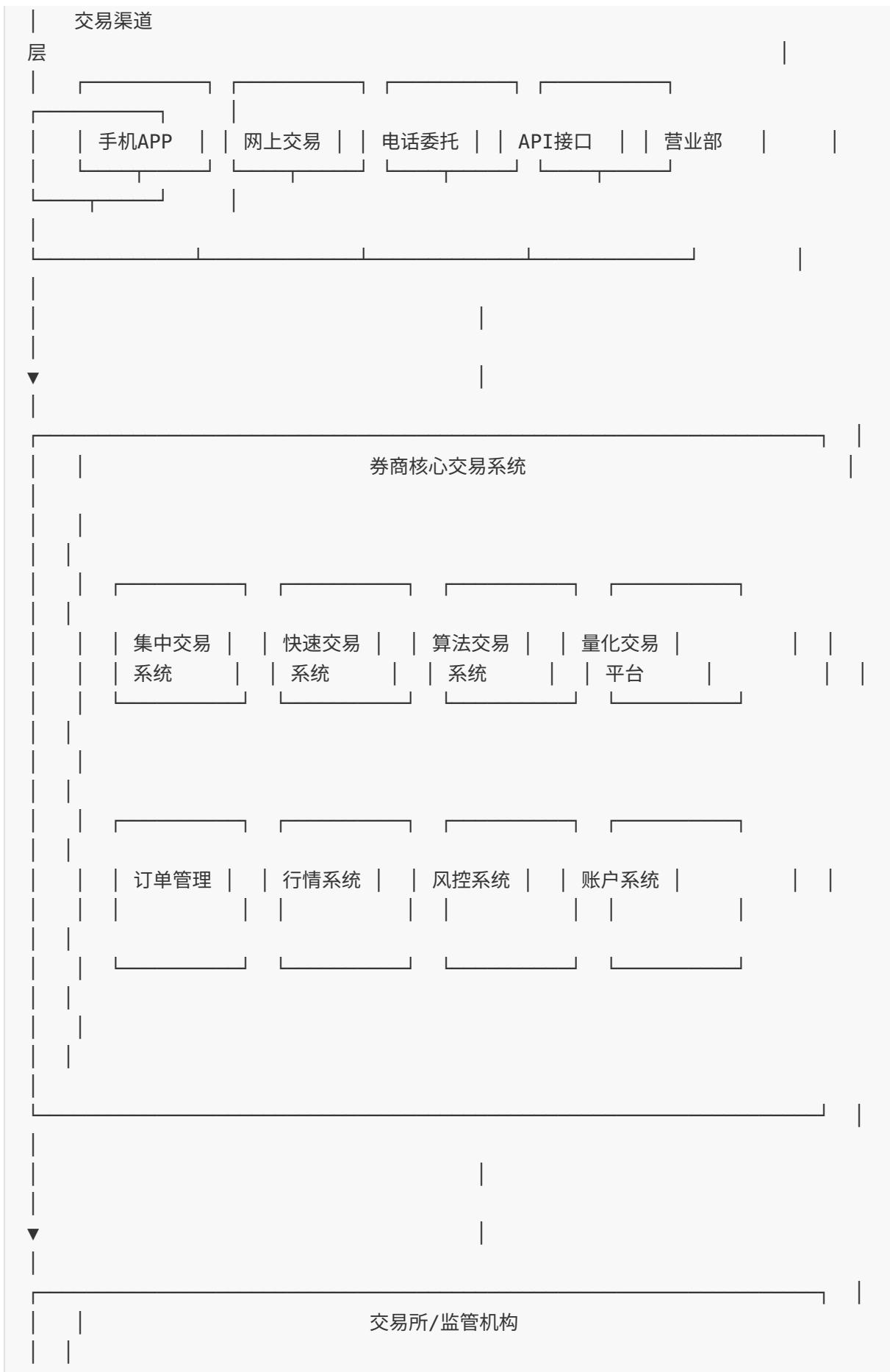
- [第10章：债券发行与承销](#)
 - [第11章：债券交易与结算](#)
 - [第五部分：工商银行业务系统](#)
 - [第12章：对公核心业务](#)
 - [第13章：企业网银与开放银行](#)
 - [第六部分：共性技术与最佳实践](#)
 - [第14章：高性能架构设计](#)
 - [第15章：数据一致性与分布式事务](#)
 - [第16章：监管合规与数据治理](#)
-

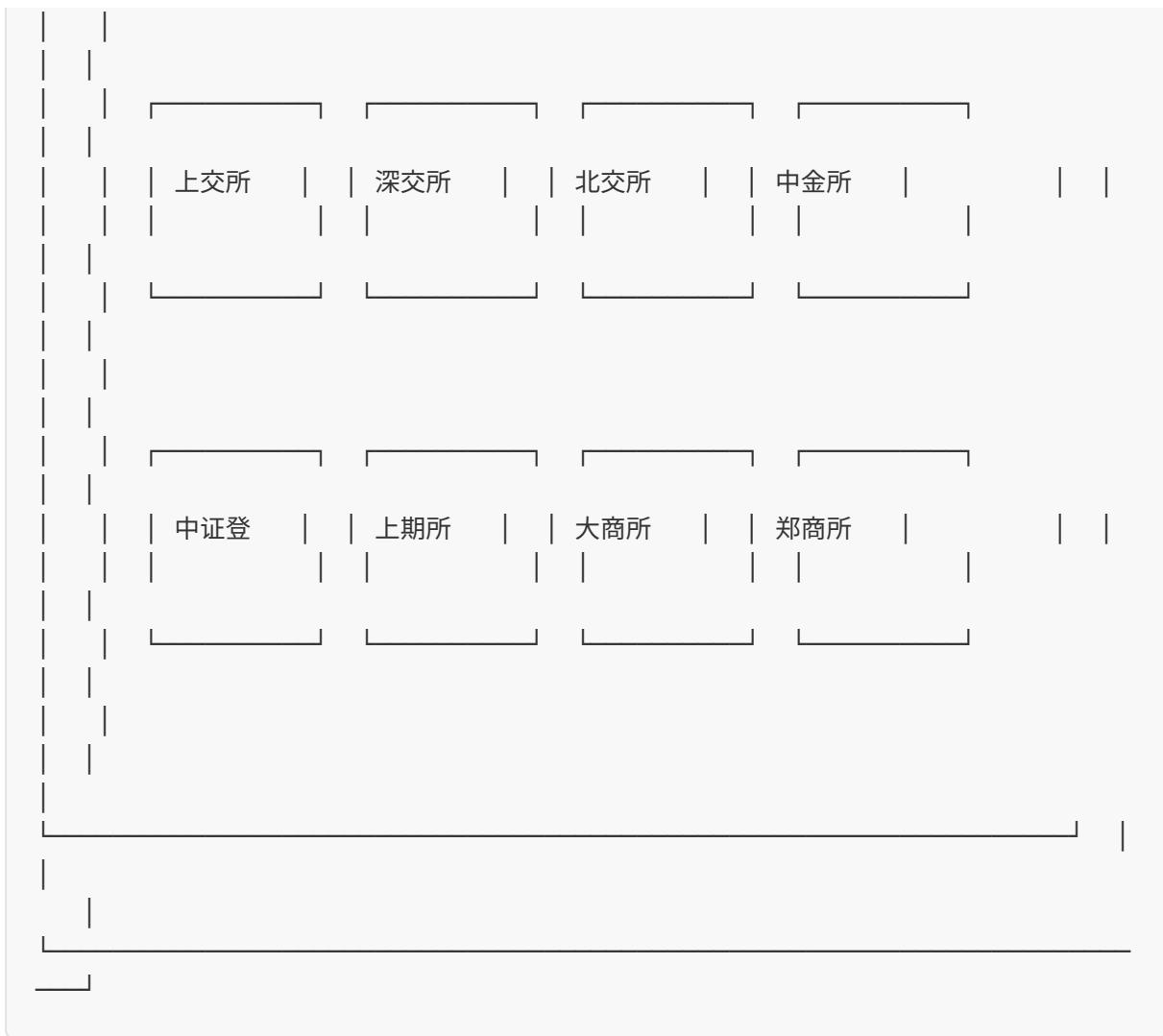
第一部分：证券业务系统

第1章：证券核心交易系统

1.1 证券交易系统生态全景



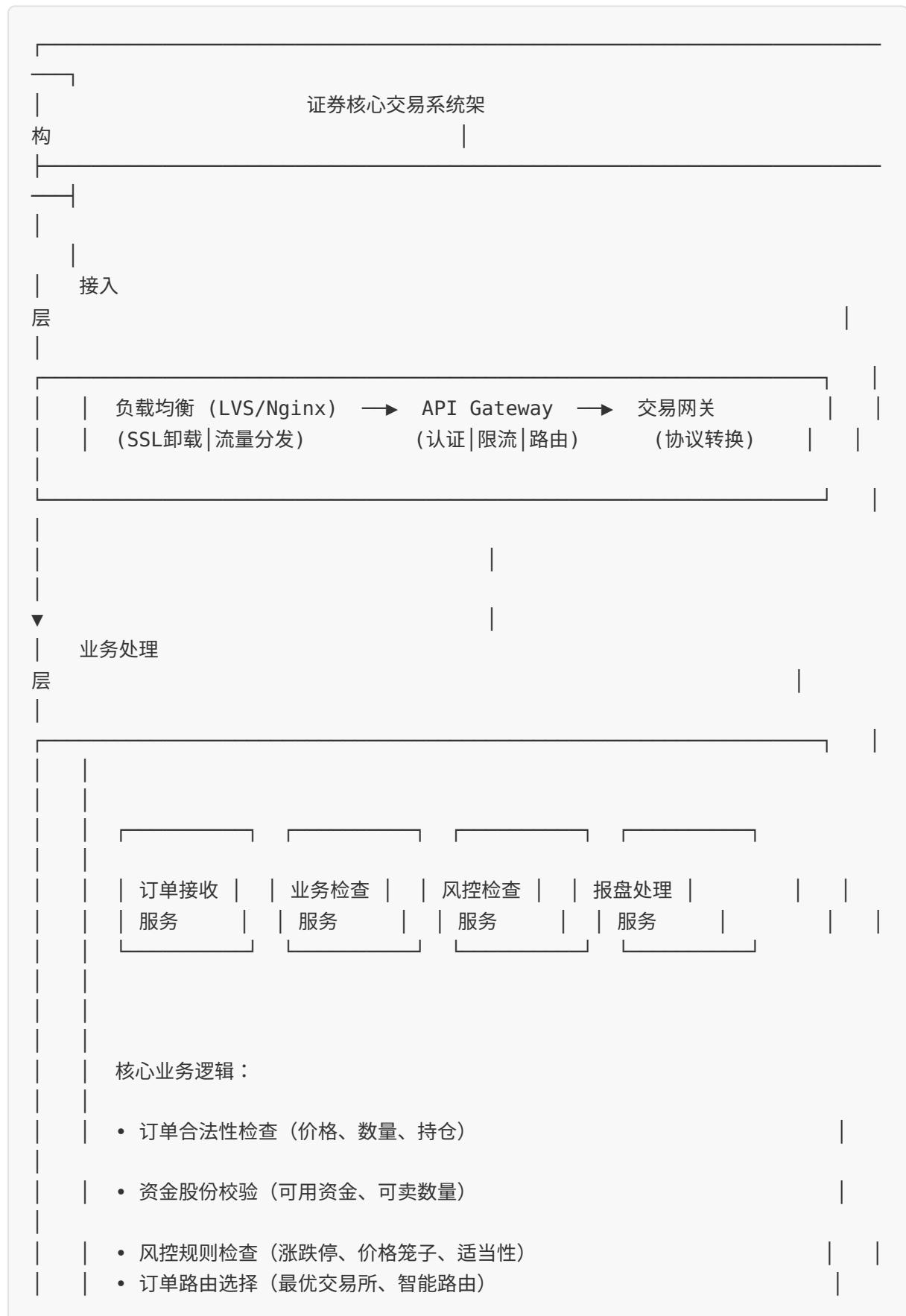


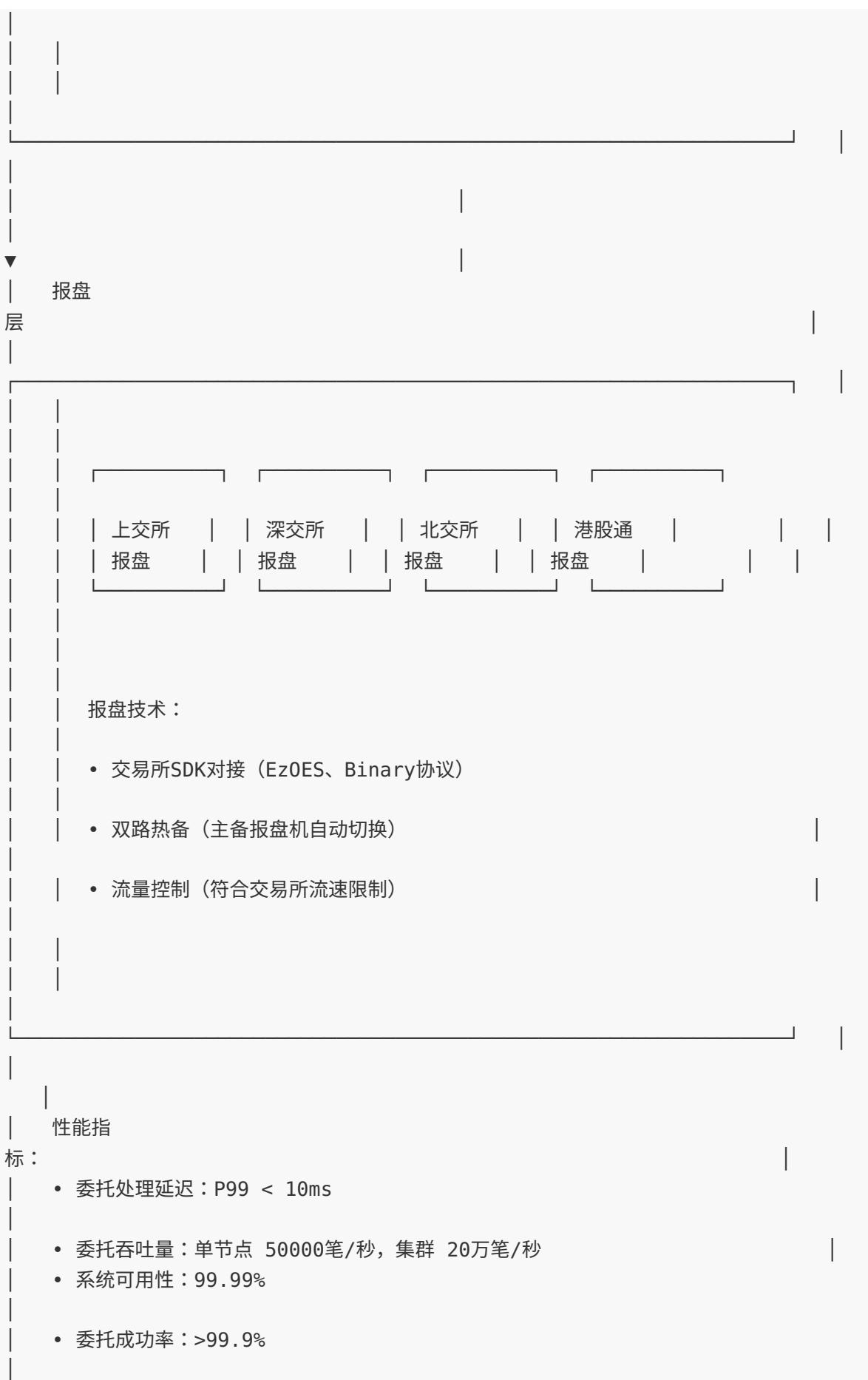


1.2 证券交易系统类型对比

系统类型	定位	延迟要求	主要用户	技术特点
集中交易系统	普通投资者交易	<100ms	散户、一般机构	高可用、大容量
快速交易系统	极速交易	<10ms	高频交易、量化	FPGA、内核旁路
算法交易系统	大单拆单执行	<50ms	机构、基金	TWAP/VWAP算法
量化交易平台	策略研发执行	<20ms	私募、自营	Python/C++、回测

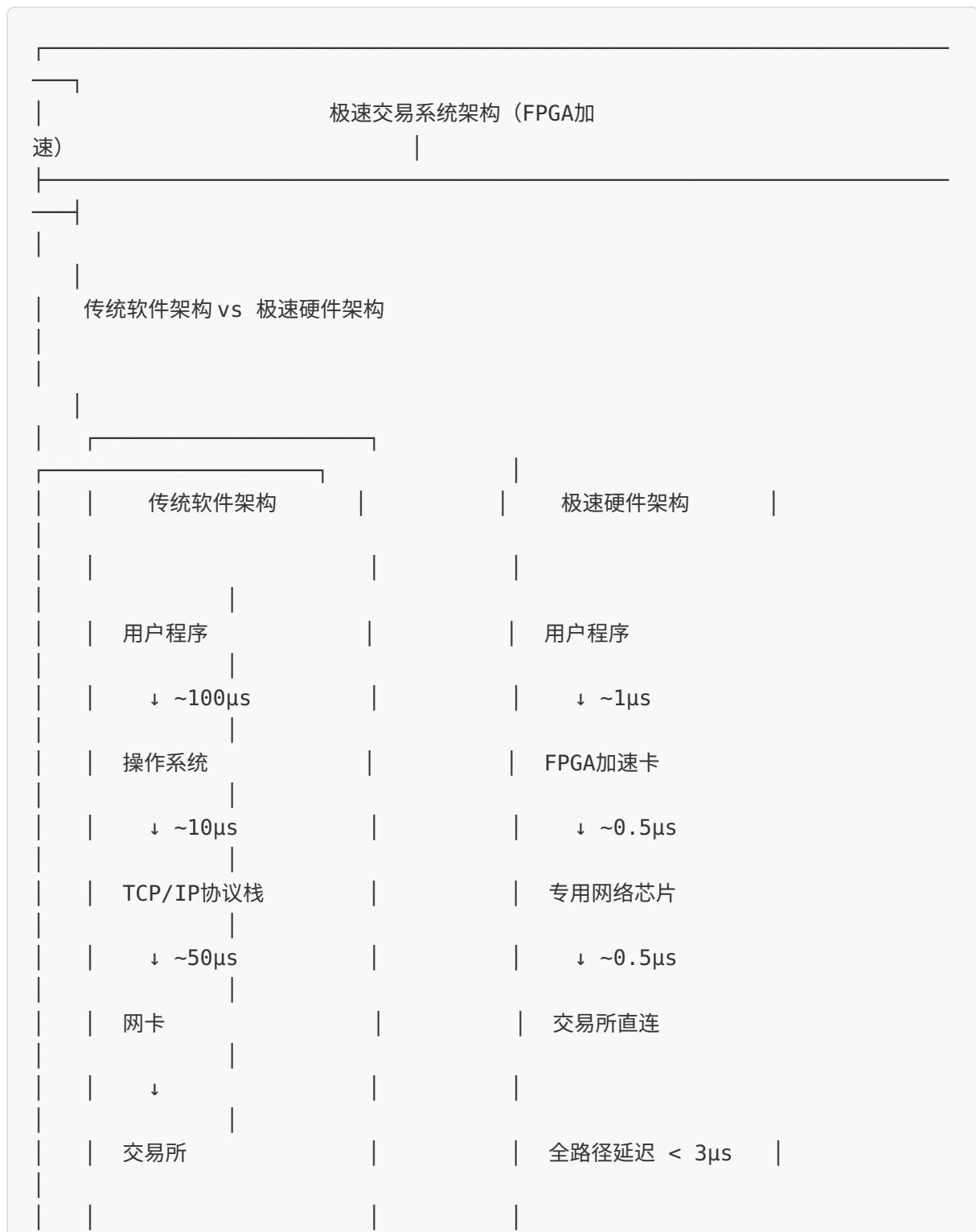
1.3 核心交易系统架构







1.4 极速交易系统架构（低延迟设计）



全路径延迟 $\sim 200\mu\text{s}$

比传统架构快100倍

极速交易技术

栈：

硬件层：

- FPGA加速卡 (Xilinx/Intel) - 网络处理、订单生成
- 智能网卡 (SmartNIC) - TCP/IP卸载、内核旁路
- 高频交易服务器 - 超频CPU、低延迟内存、NVMe SSD

网络层：

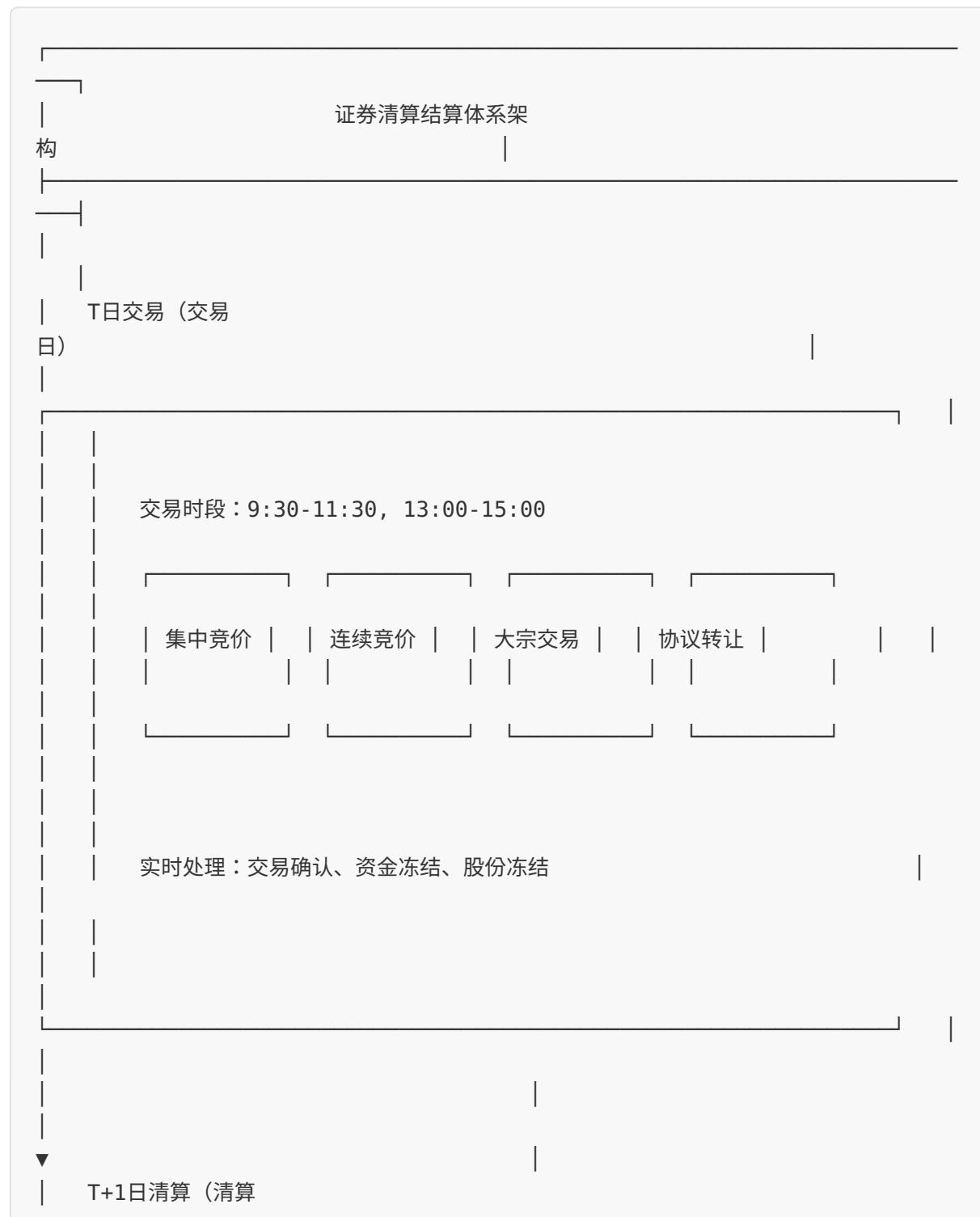
- kernel bypass (DPDK/RDMA) - 绕过操作系统内核
- 专线接入 - 交易所托管机房同城部署
- 10G/25G/100G以太网 - 低延迟交换机和网卡

软件层：

- C++高性能计算 - 无锁队列、内存池、零拷贝
- 实时操作系统 - 禁用中断、CPU亲和性绑定

第2章：证券清算结算系统

2.1 清算结算体系架构



日)

日终清算流程：

1. 交易数据接收（交易所收盘文件）

↓

2. 清算处理（净额清算、逐笔清算）

↓

3. 资金计算（应收应付、费用计算）

↓

4. 股份过户（过户登记、权益处理）

↓

5. 资金交收（银证转账、央行支付）

参与

方

券商

清算

中证登

结算

交易所

交易数据

银行

资金存管

央行

大额支付



2.2 清算系统核心模块

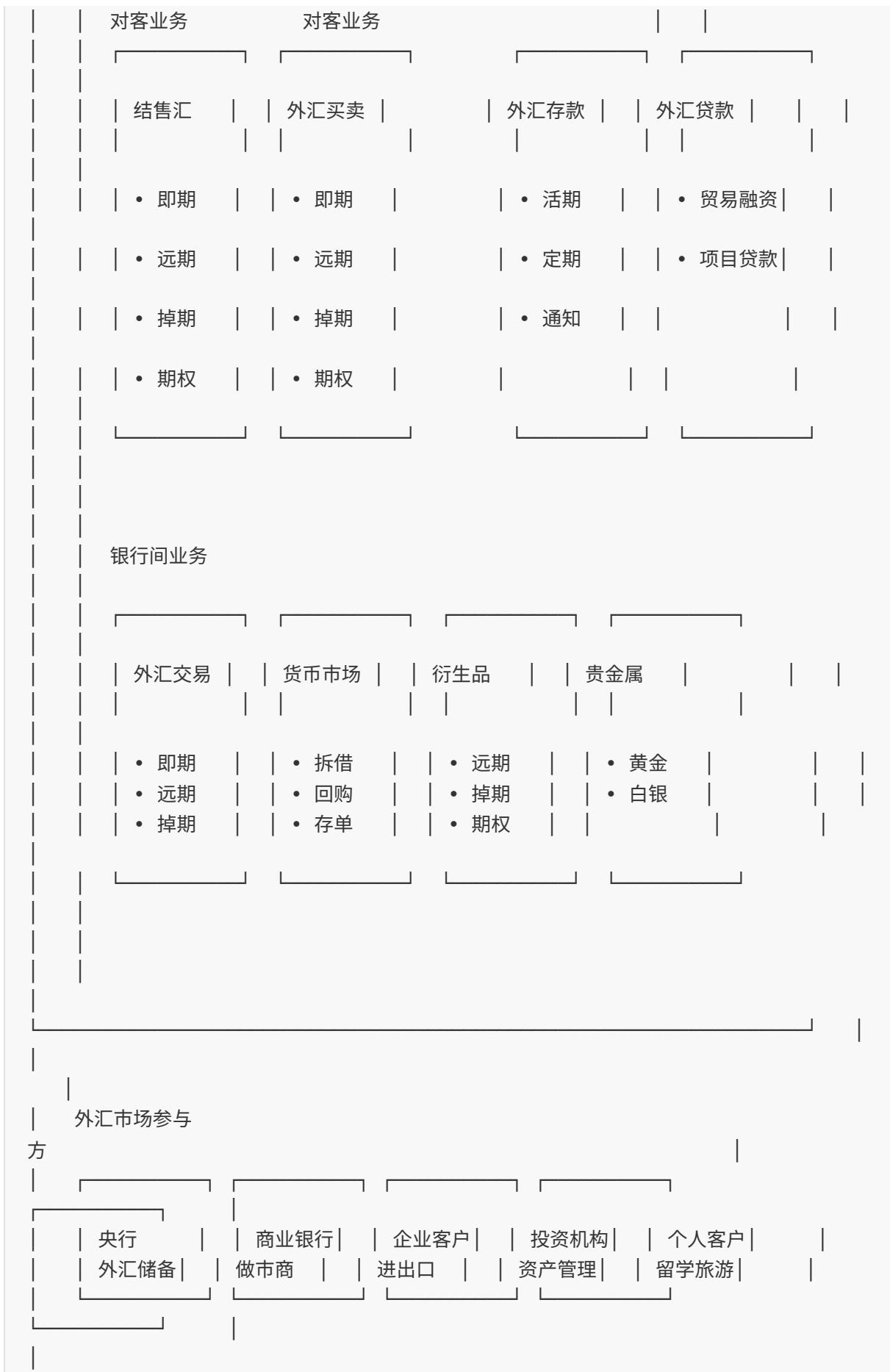
模块	功能	关键指标
数据接收	接收交易所成交数据、行情数据	完整性校验、时效性
清算处理	买卖净额计算、费用计算	计算准确性、处理时效
资金交收	资金划拨、银证转账	资金安全、到账时效
股份交收	过户登记、权益派发	股权准确性、及时性
对账处理	与交易所、银行、中证登对账	对账一致性、差错处理

第二部分：外汇业务系统

第4章：外汇交易系统

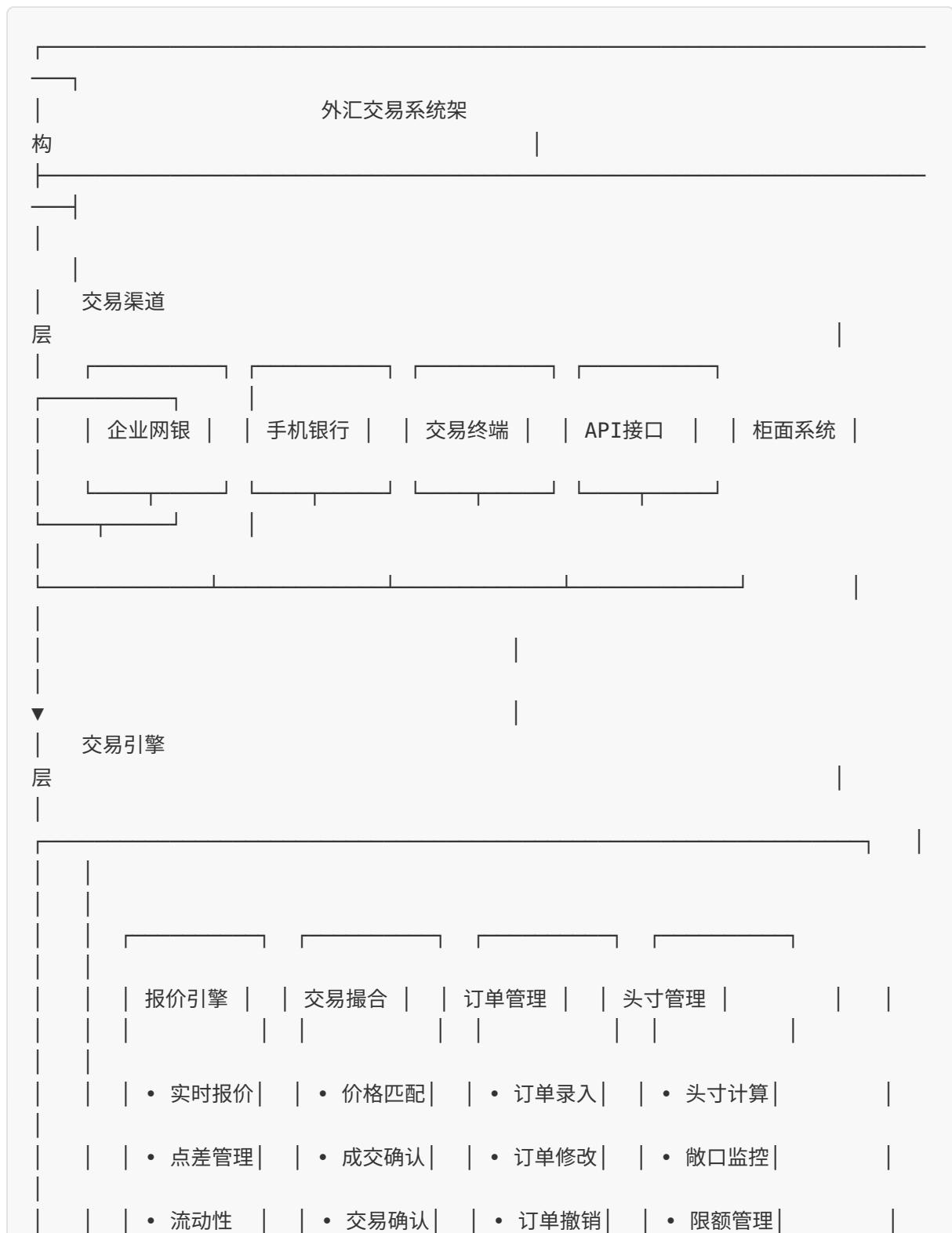
4.1 外汇业务全景

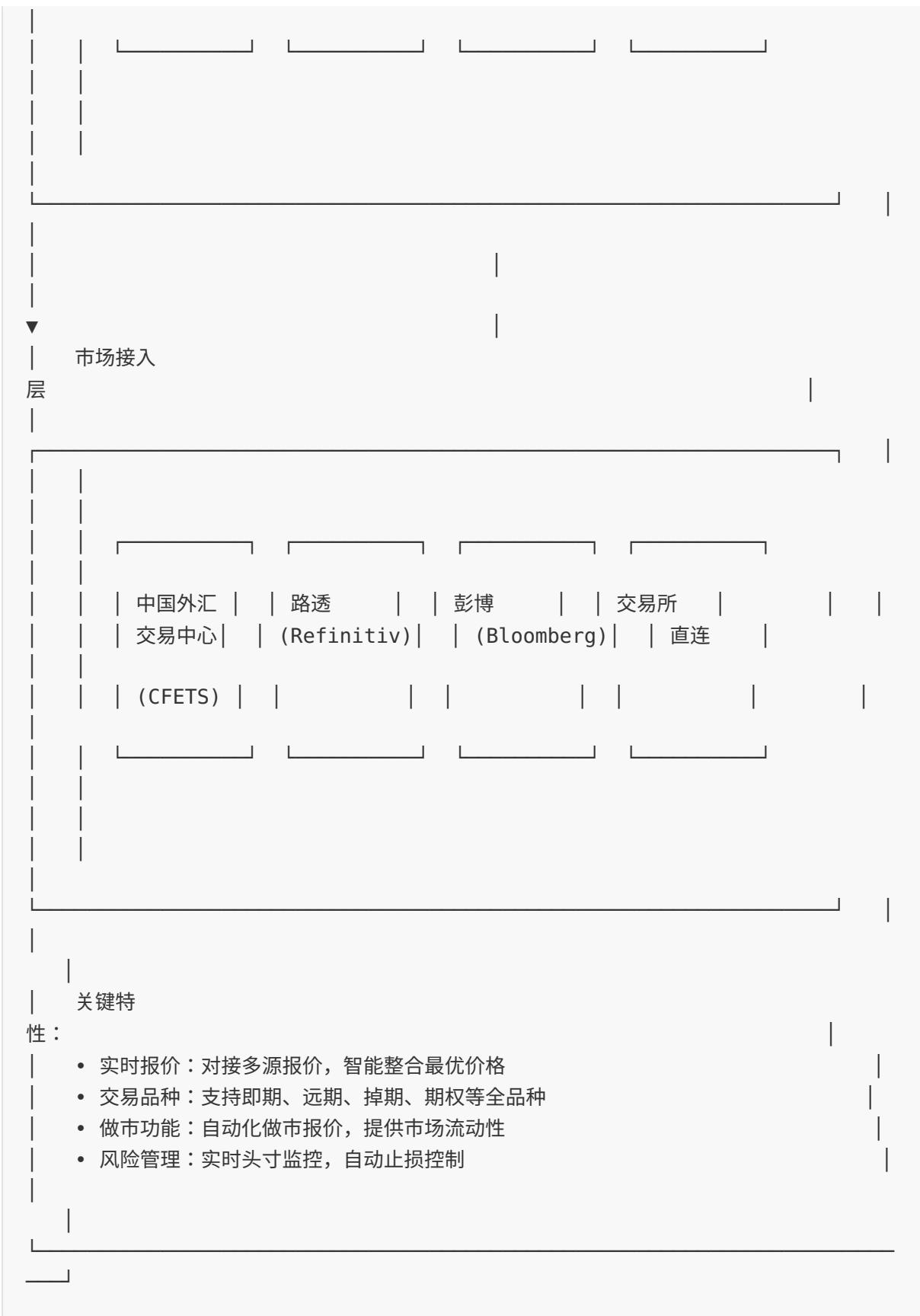






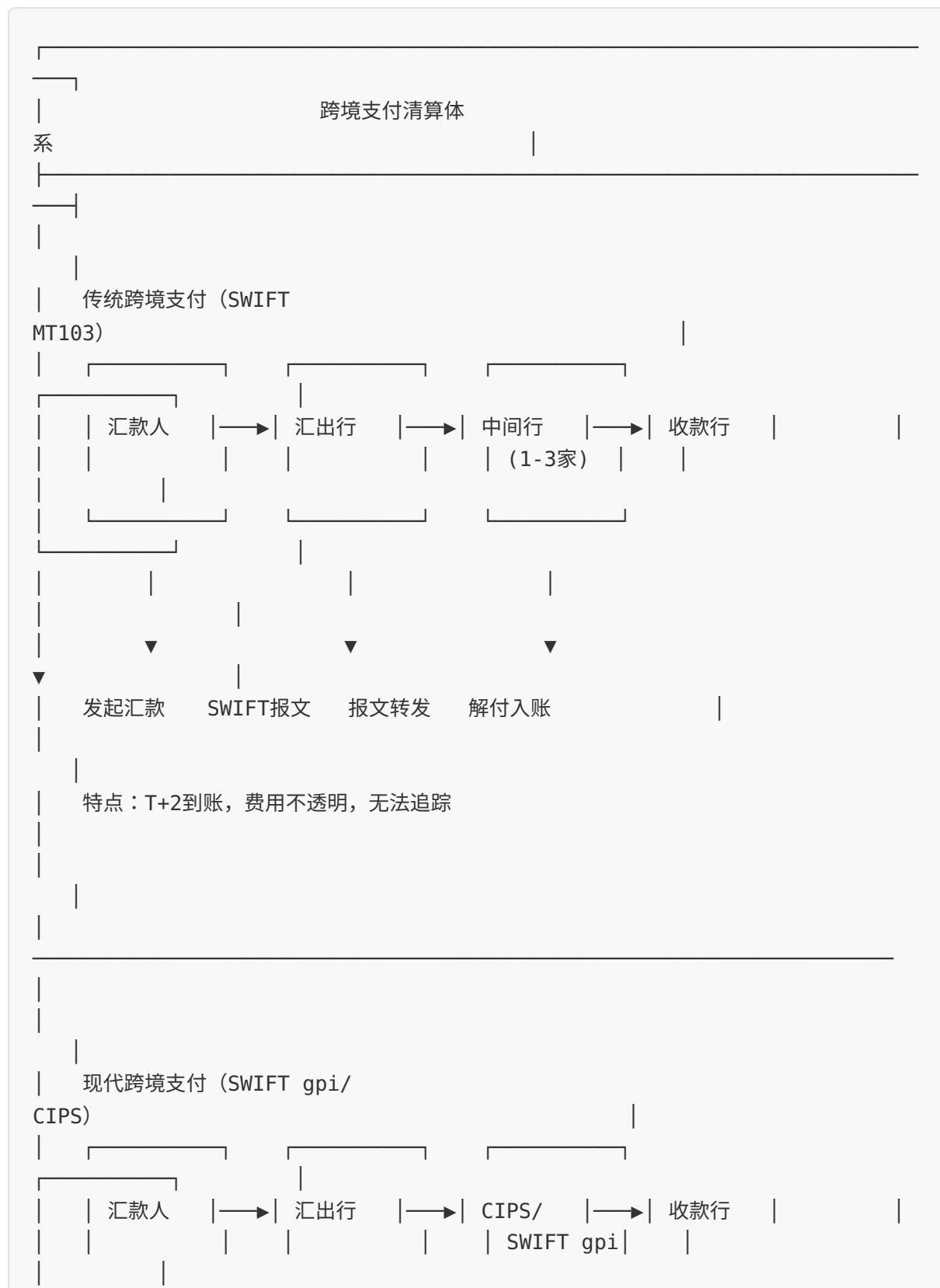
4.2 外汇交易系统架构





第5章：跨境支付与清算

5.1 跨境支付体系





5.2 CIPS（人民币跨境支付系统）架构

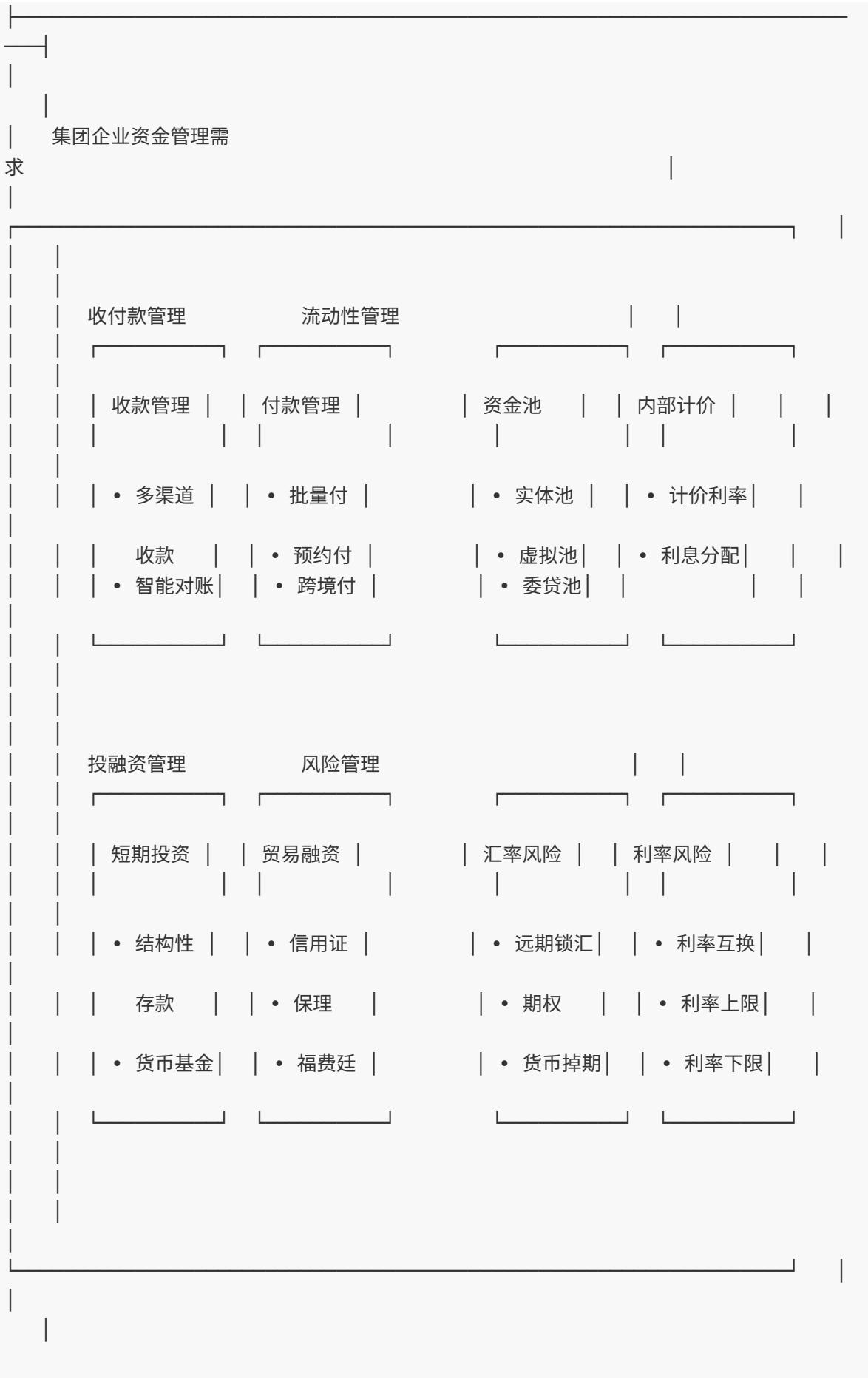
组件	功能	特点
CIPS直参行	直接接入CIPS系统	实时全额结算，需央行审批
CIPS间参行	通过直参行间接接入	成本较低，结算有延迟
HVPS（大额支付）	提供央行账户资金结算	实时全额，最终性
FXCC（外汇交易）	外汇交易结算	同步交收（PVP）

第三部分：交易银行业务系统

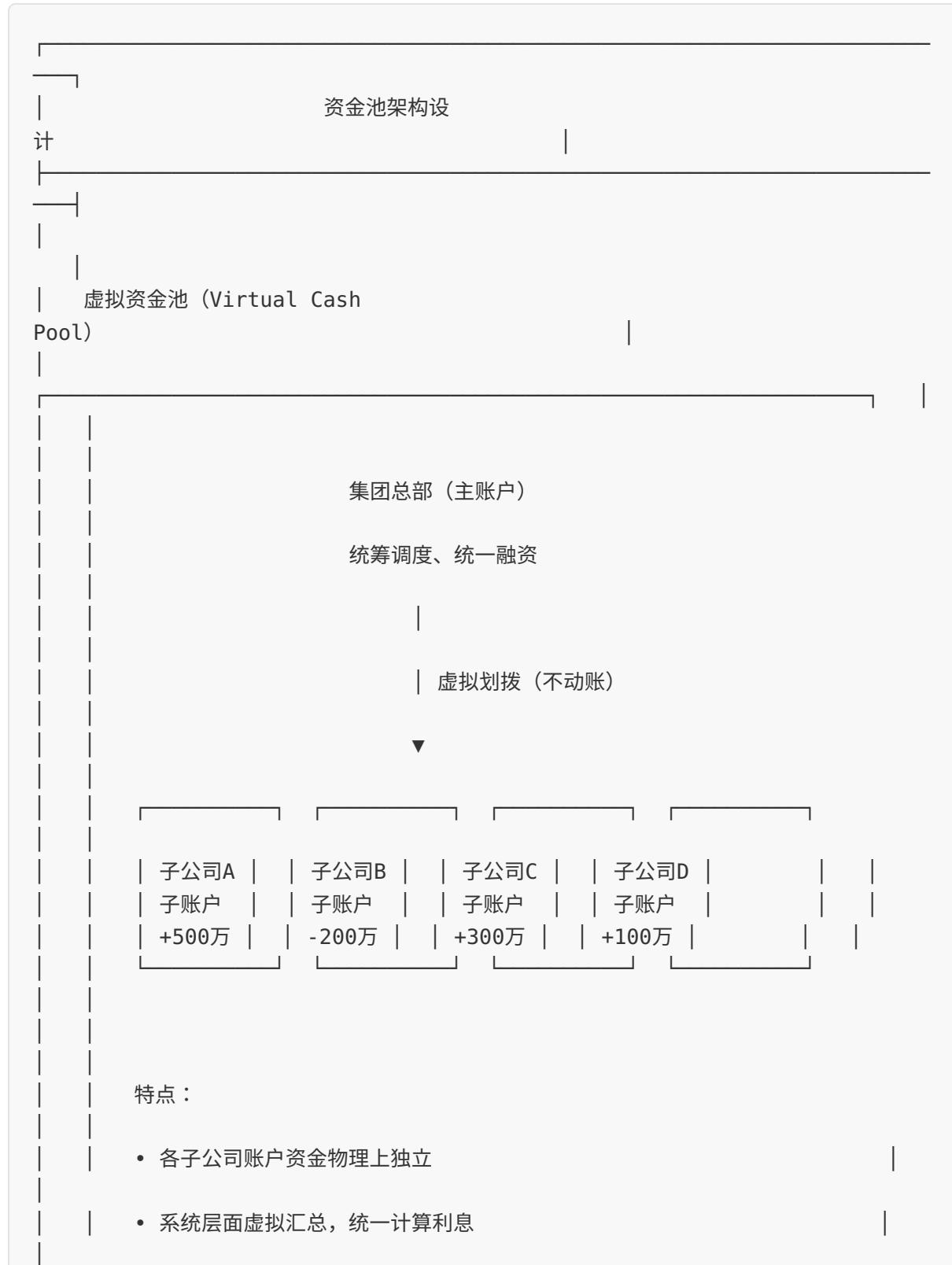
第7章：现金管理系统

7.1 现金管理业务全景





7.2 资金池架构设计



- 透支子公司自动占用总部额度

实体资金池 (Physical Cash Pool)

集团总部 (主账户)

资金 physically 归集

| 定时/实时上划



子公司A	子公司B	子公司C	子公司D
子账户	子账户	子账户	子账户
余额≈0	余额≈0	余额≈0	余额≈0

特点：

- 资金物理上划至主账户
- 子公司账户保持零余额或目标余额
- 需要委贷或资金下拨实现资金调配

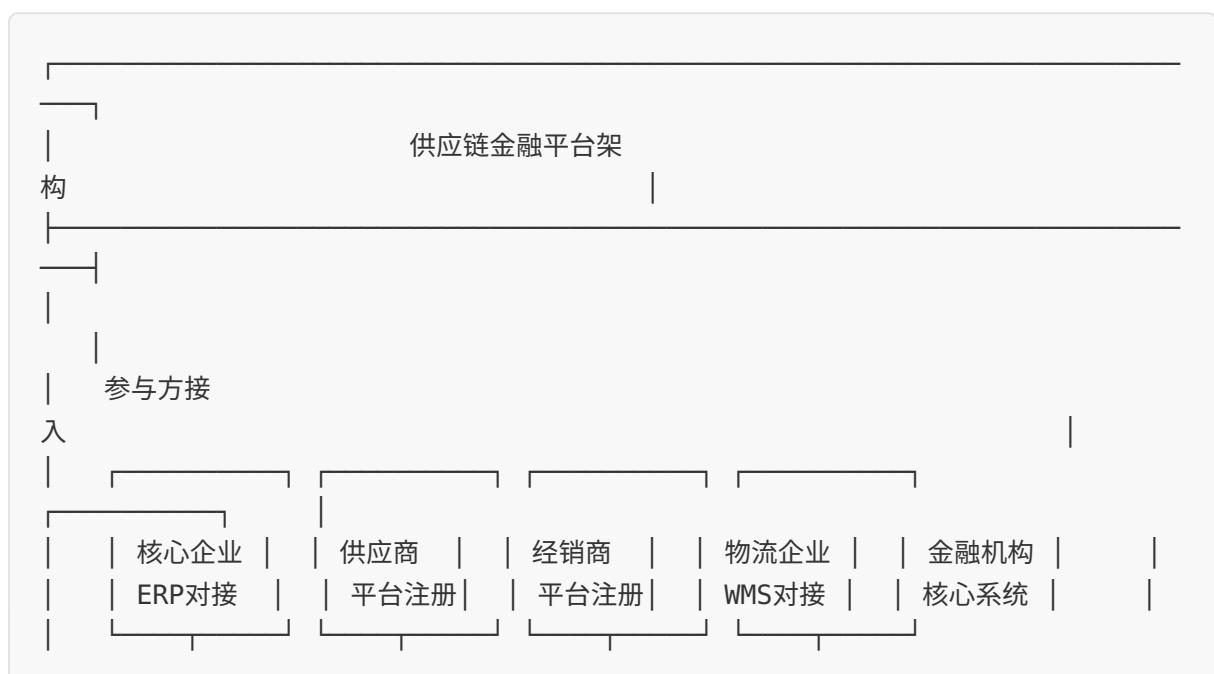


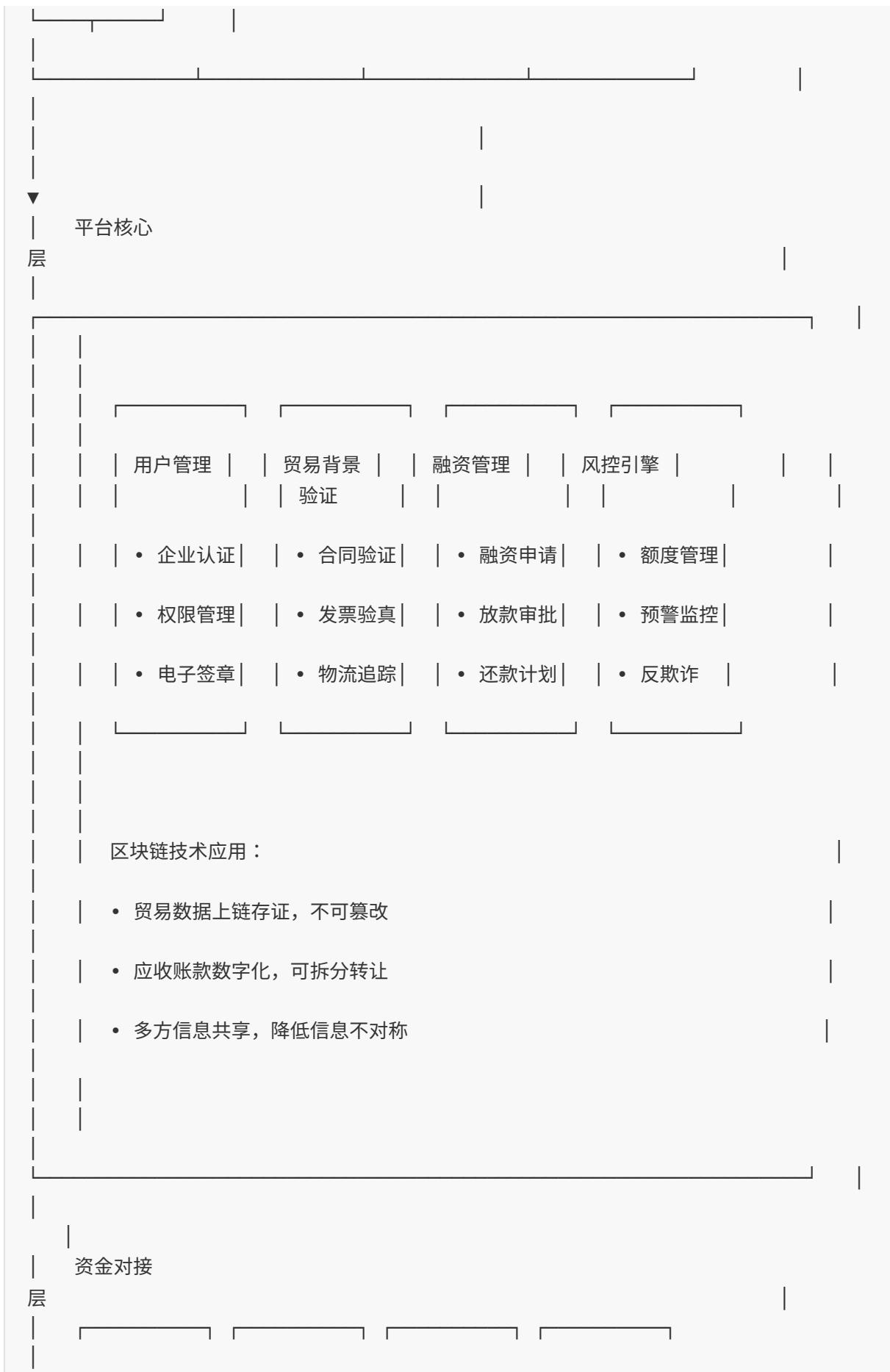
第8章：供应链金融系统

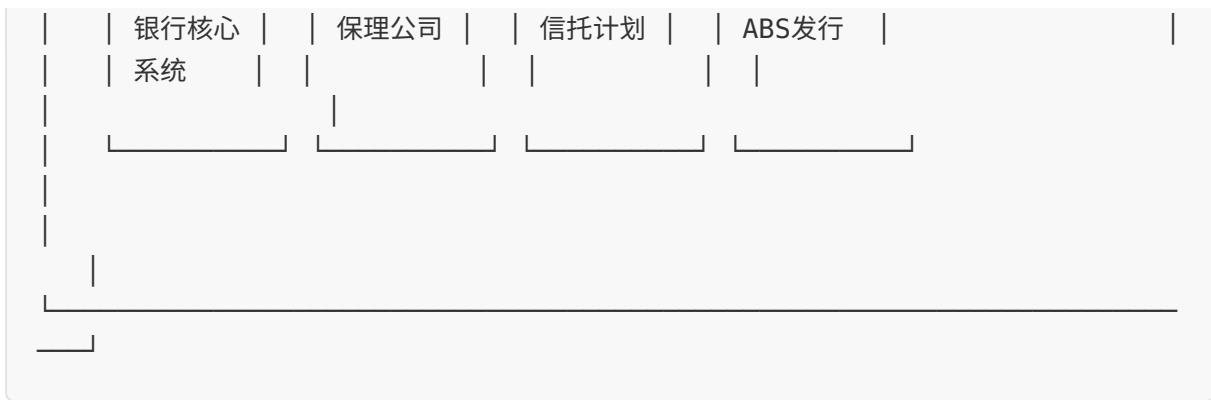
8.1 供应链金融模式

模式	参与方	核心企业	融资对象	风控要点
应收账款融资	供应商、核心企业、银行	信用背书	供应商	应收账款确权、回款账户监管
预付款融资	经销商、核心企业、银行	货物监管	经销商	货物监管、跌价补偿
存货融资	借款企业、仓储方、银行	第三方监管	借款企业	货物监管、动态盯市
保理	供应商、保理商、核心企业	信用背书	供应商	应收账款转让通知

8.2 供应链金融平台架构



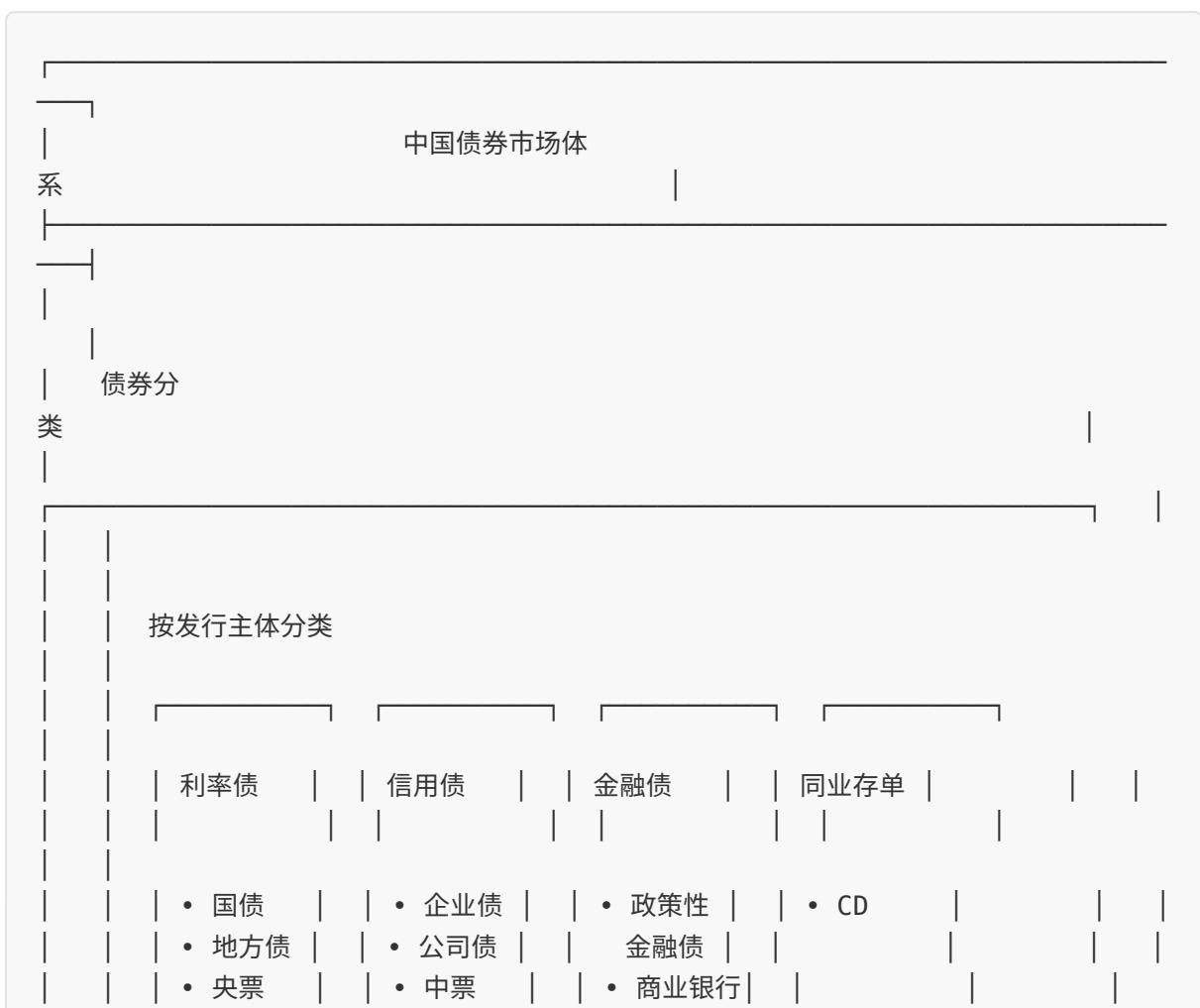


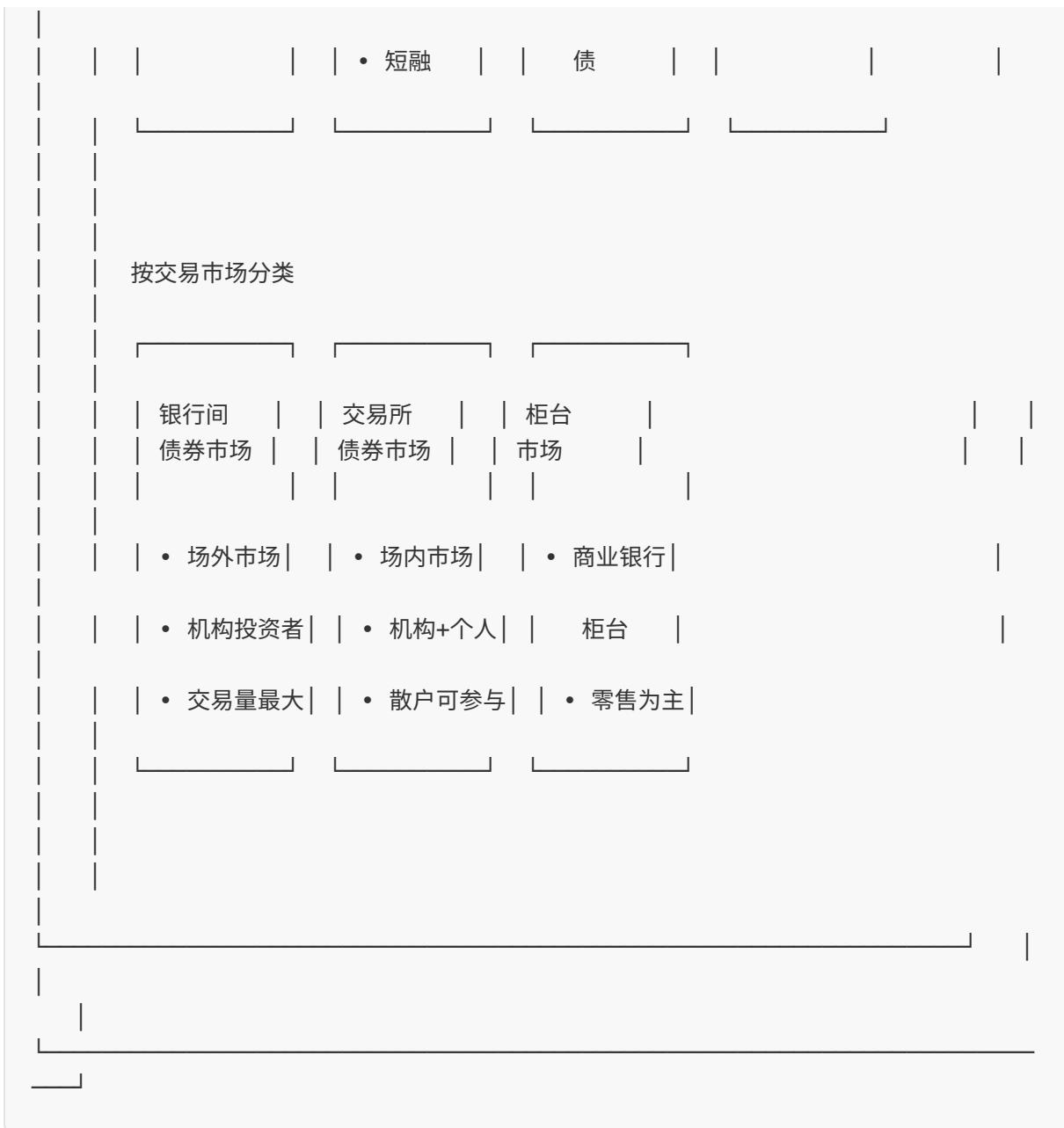


第四部分：债券业务系统

第10章：债券发行与承销

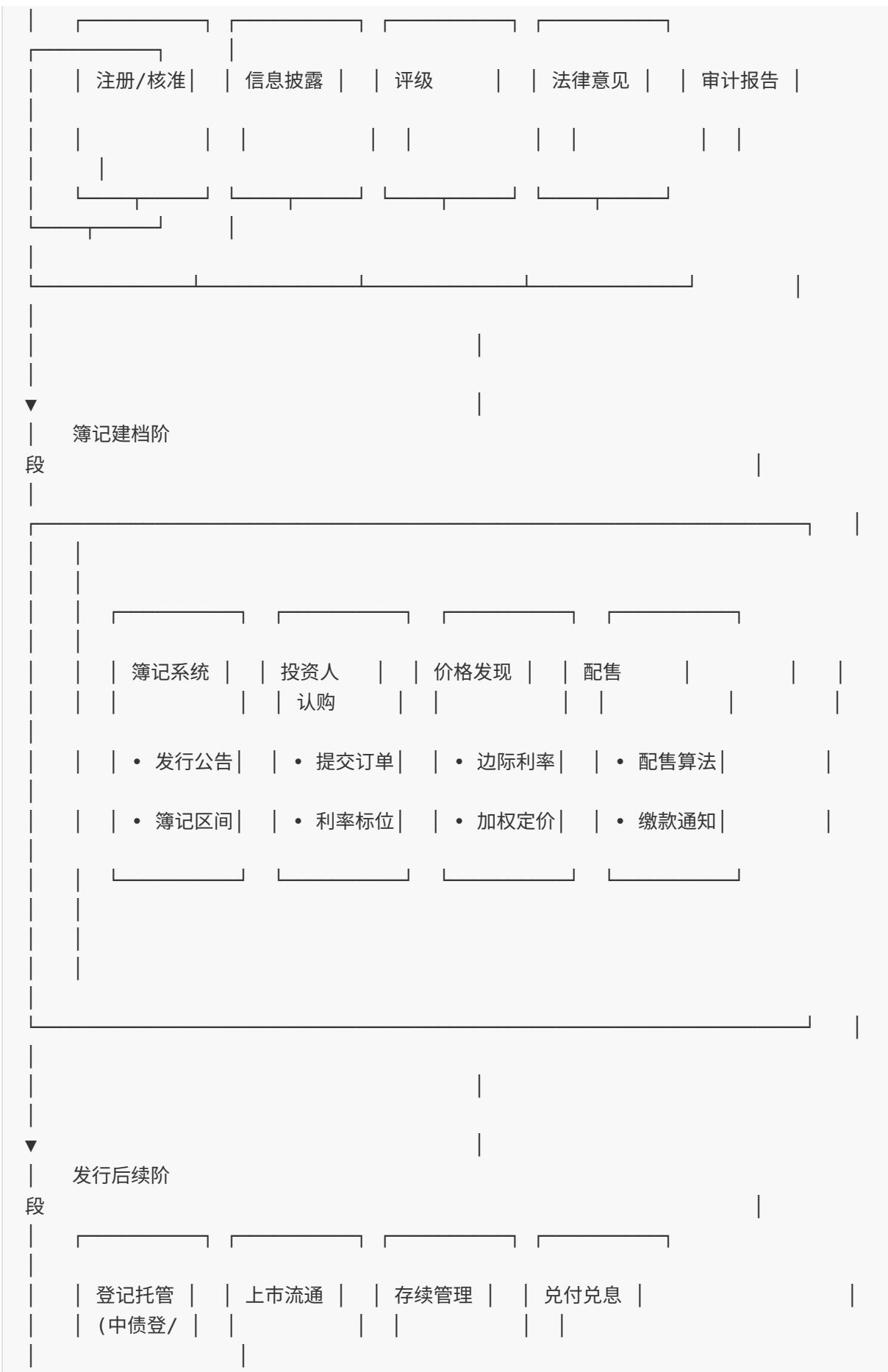
10.1 债券市场体系

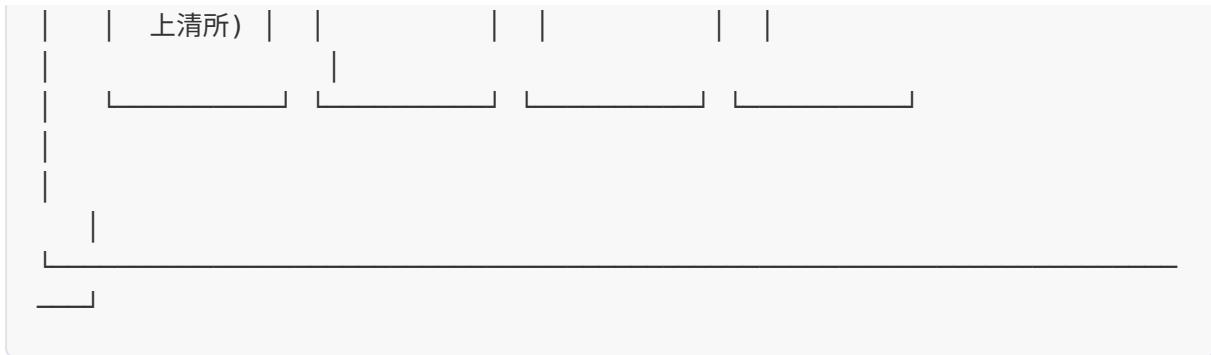




10.2 债券发行系统架构



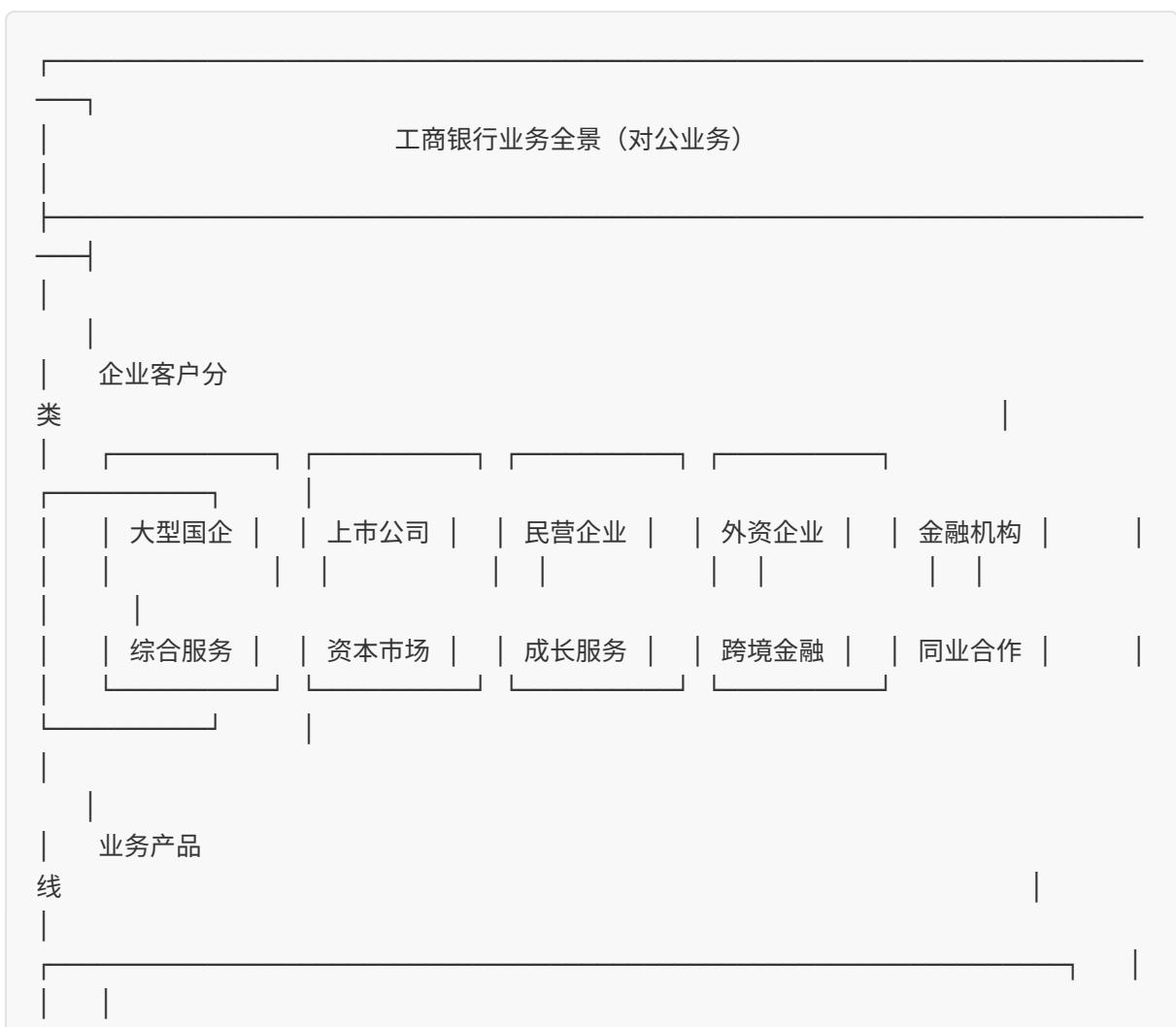


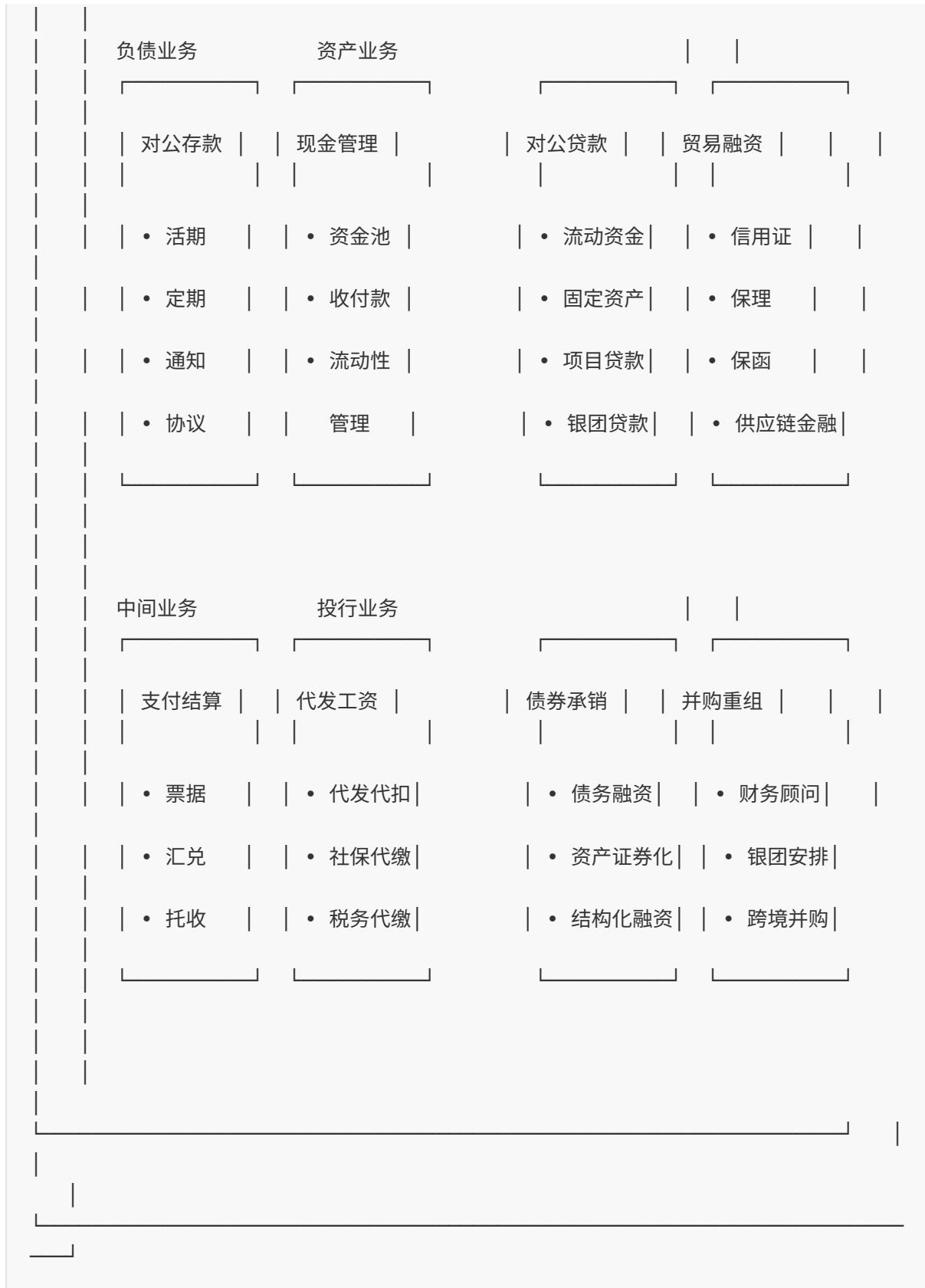


第五部分：工商银行业务系统

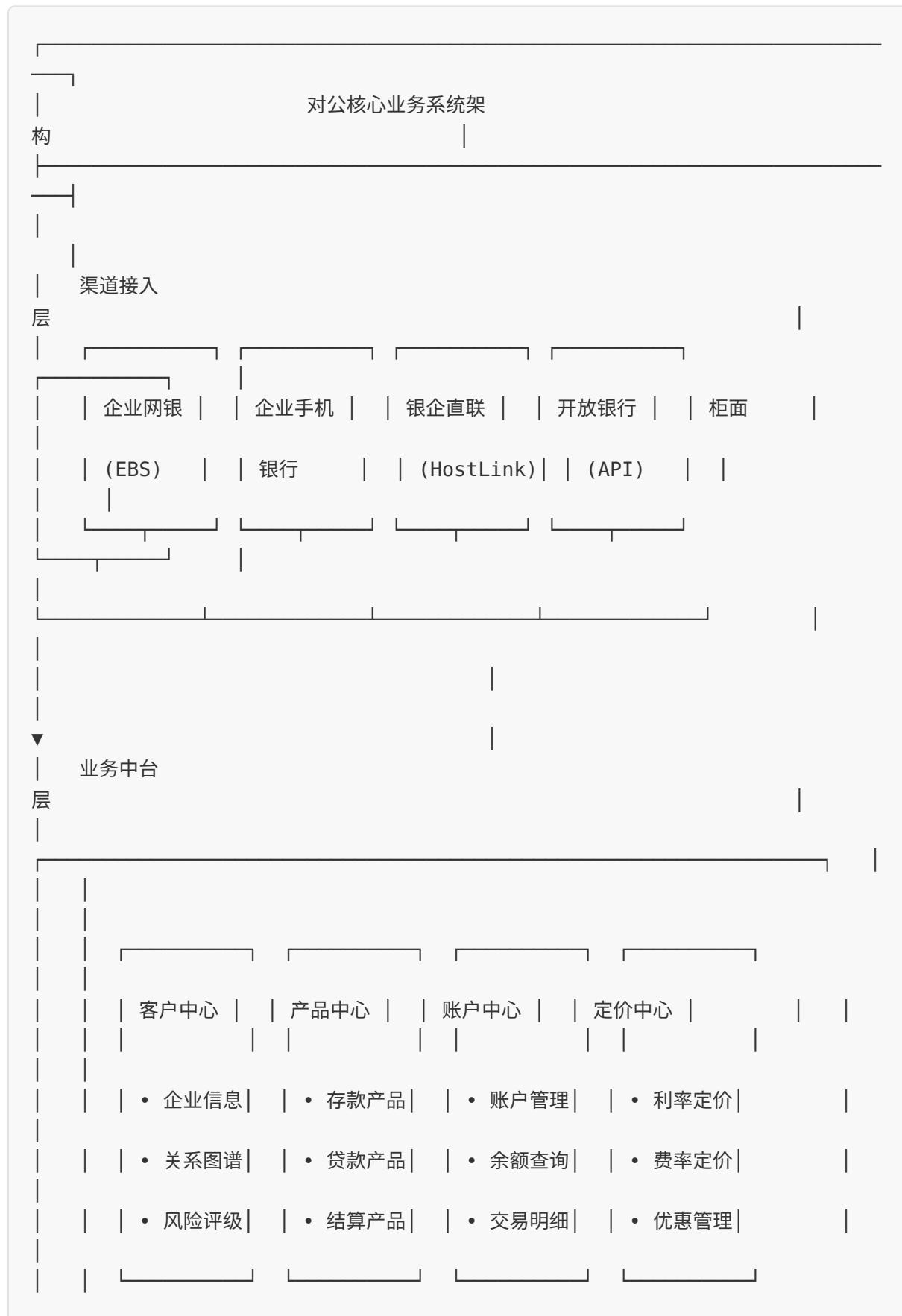
第12章：对公核心业务

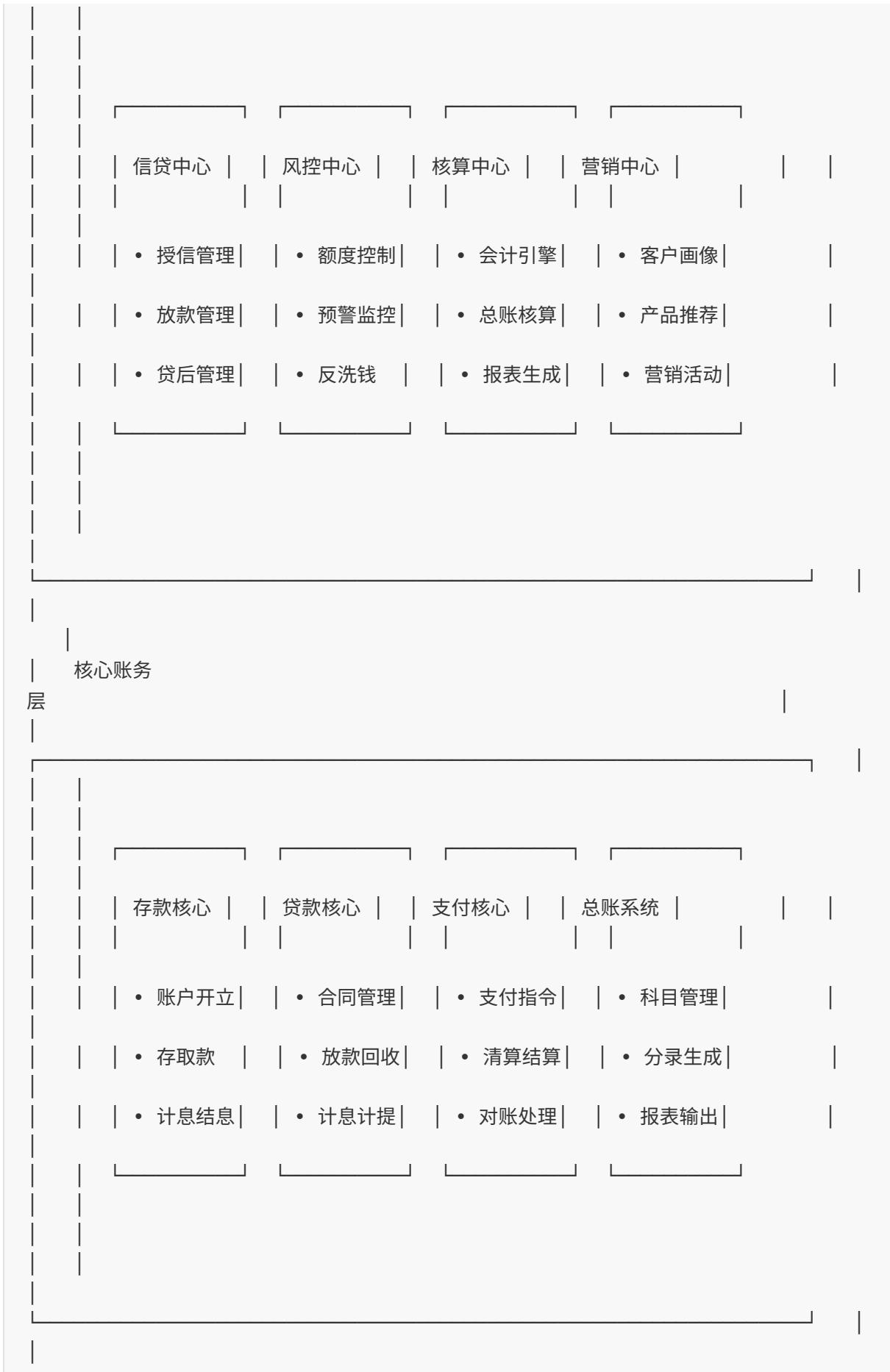
12.1 工商银行业务全景





12.2 对公核心系统架构



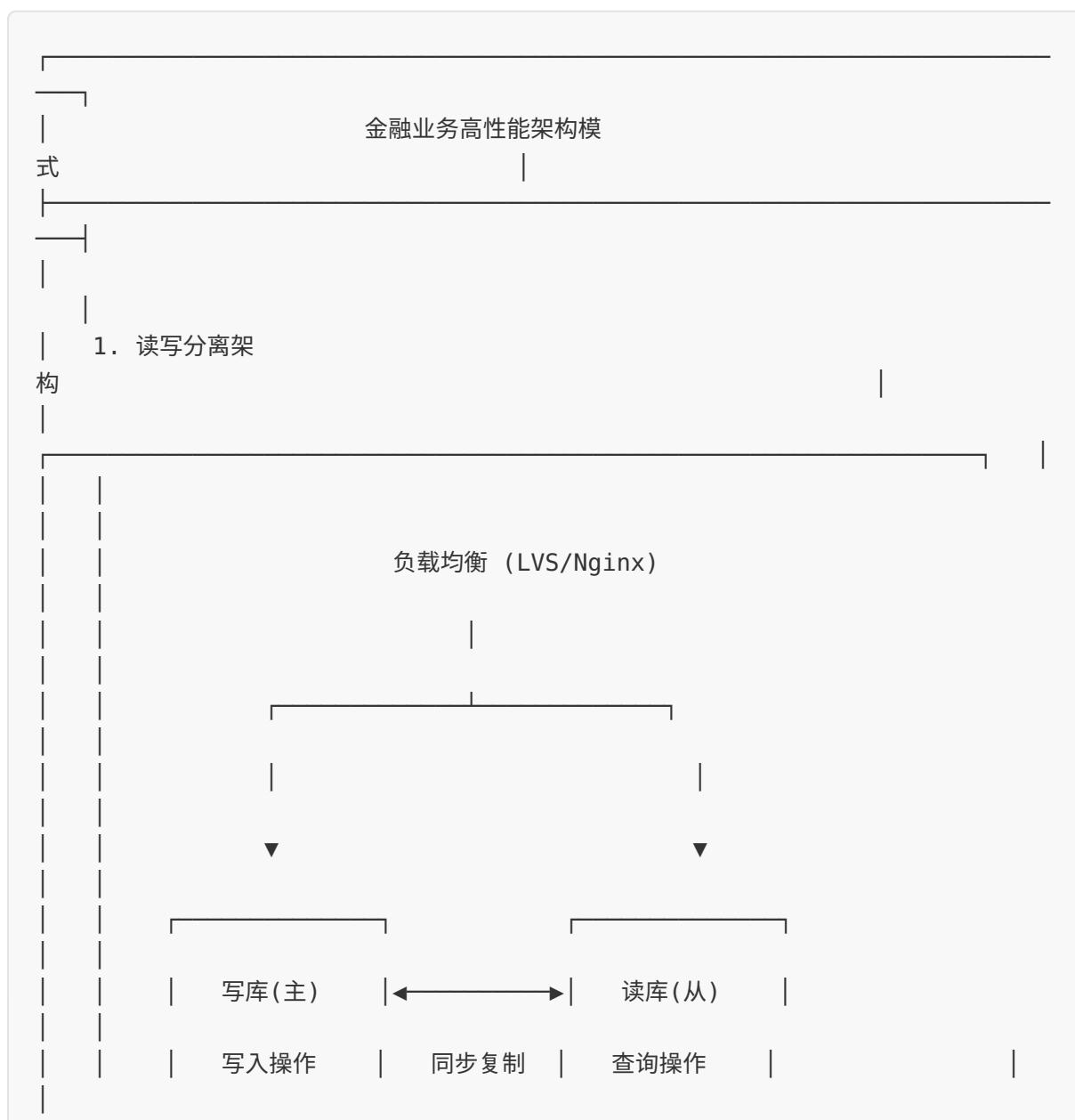




第六部分：共性技术与最佳实践

第14章：高性能架构设计

14.1 高性能架构模式



适用场景：证券行情查询、银行余额查询等读多写少场景

2. 分库分表架构

分片策略：

- 水平分片：按客户ID哈希分片，数据均匀分布
- 垂直分片：按业务域分片，如客户信息、交易记录分离
- 时间分片：按时间维度分片，如按月分区

路由层：ShardingSphere / MyCAT

→ DB1 (客户ID: 0-9999)

→ DB2 (客户ID: 10000-19999)

→ DB3 (客户ID: 20000-29999)

→ DB4 (客户ID: 30000+)

适用场景：海量客户数据、历史交易数据存储

3. 缓存架构

多级缓存：

L1：本地缓存（Caffeine/Guava）

- 命中率：90%+
- 延迟： $< 1\text{ms}$

L2：分布式缓存（Redis Cluster）

- 命中率：95%+
- 延迟： $< 5\text{ms}$

L3：数据库（TiDB/OceanBase）

- 最终兜底
- 延迟： $< 10\text{ms}$

适用场景：证券行情、外汇牌价、客户信息等热点数据

第15章：数据一致性与分布式事务

15.1 金融业务一致性要求

业务场景	一致性要求	实现方案	技术选型
证券成交	强一致	2PC/TCC	自研事务框架
银行转账	强一致	TCC + 对账	Seata/Dromara
外汇交易	最终一致	Saga + 补偿	消息队列
支付清算	最终一致	本地消息表	RocketMQ事务消息
积分扣减	最终一致	异步补偿	Redis + MQ

15.2 分布式事务实现

```
/**
 * 银行转账TCC实现示例
 */
@Service
public class TransferService {

    @Autowired
    private AccountTccAction accountTccAction;

    /**
     * 转账入口
     */
    @GlobalTransactional(name = "transfer", rollbackFor =
Exception.class)
    public void transfer(String fromAccount, String toAccount,
BigDecimal amount) {
        // Try阶段：预留资源
        boolean tryResult = accountTccAction.tryTransfer(fromAccount,
toAccount, amount);
        if (!tryResult) {
```

```

        throw new BusinessException("转账预留资源失败");
    }

    // 其他业务逻辑...

    // Confirm阶段：自动提交（由TC协调）
    // Cancel阶段：异常时回滚（由TC协调）
}
}

/**
 * TCC接口定义
 */
@LocalTCC
public interface AccountTccAction {

    @TwoPhaseBusinessAction(name = "transferAction",
                           tryMethod = "tryTransfer",
                           confirmMethod = "commitTransfer",
                           rollbackMethod = "rollbackTransfer")
    boolean tryTransfer(@BusinessActionContextParameter(paramName =
"from") String fromAccount,
                     @BusinessActionContextParameter(paramName =
"to") String toAccount,
                     @BusinessActionContextParameter(paramName =
"amount") BigDecimal amount);

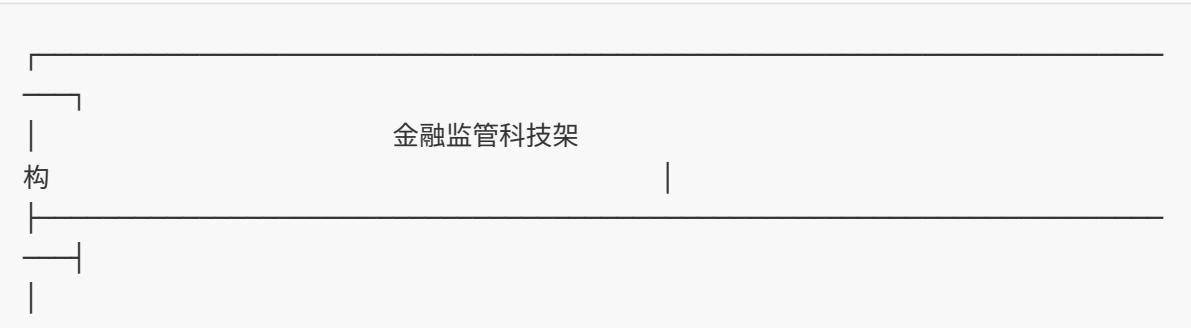
    boolean commitTransfer(BusinessActionContext context);

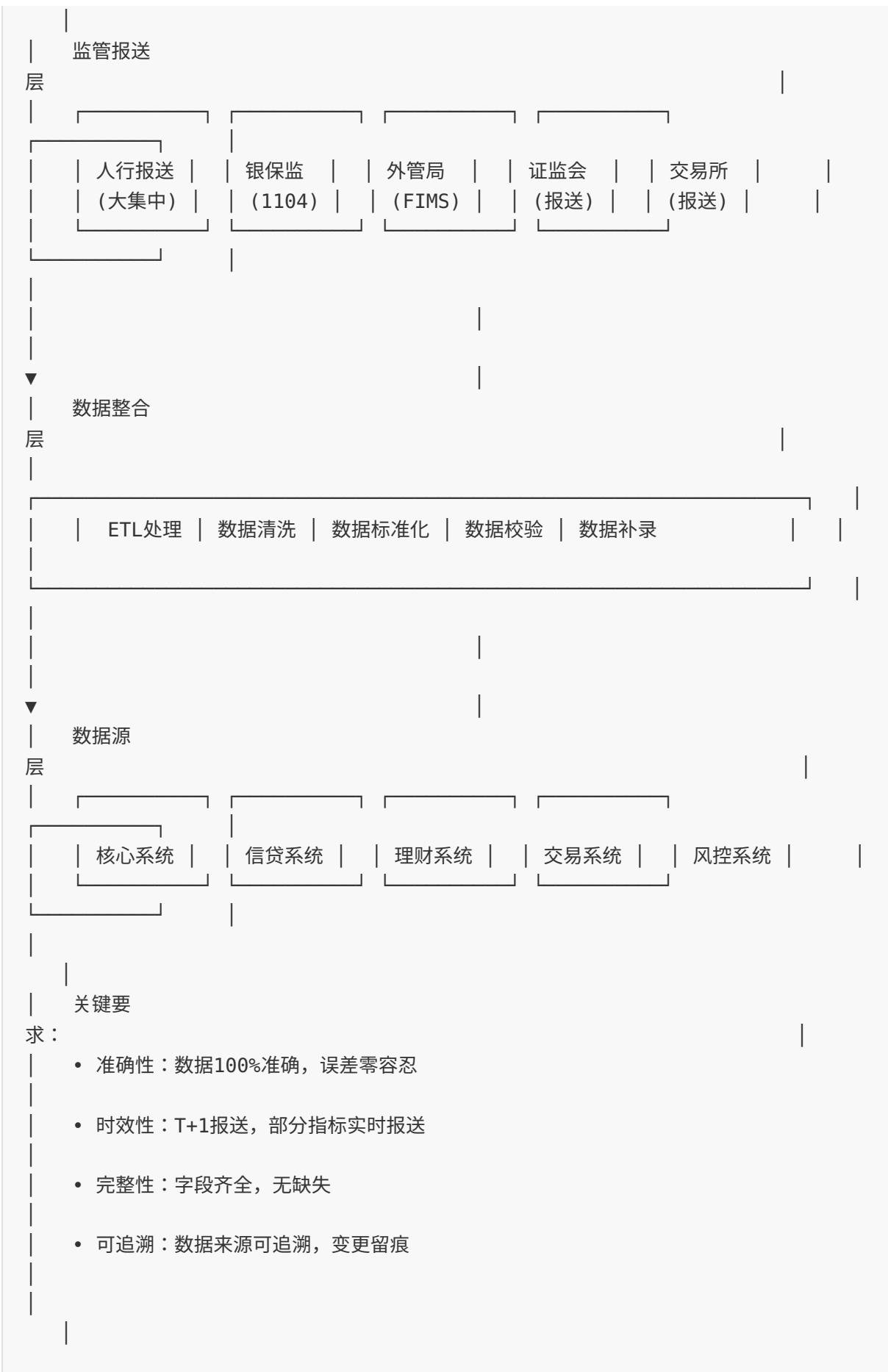
    boolean rollbackTransfer(BusinessActionContext context);
}

```

第16章：监管合规与数据治理

16.1 金融监管科技 (RegTech)







16.2 数据治理框架

治理领域	核心任务	实施要点
数据标准	统一数据定义、编码规范	企业级数据字典、标准管理流程
数据质量	准确性、完整性、一致性	质量规则、监控告警、问题闭环
元数据管理	数据血缘、影响分析	自动采集、可视化展示
数据安全	分类分级、访问控制	敏感识别、脱敏加密、审计追溯
生命周期	归档、清理、销毁	分级存储、自动归档

总结

本文档全面解析了五大专业金融业务领域的系统架构和开发最佳实践：

证券业务

- 核心交易系统：高并发、低延迟、高可用
- 极速交易：FPGA加速、kernel bypass
- 清算结算：T+1清算、资金交收

外汇业务

- 外汇交易：实时报价、多币种、做市功能
- 跨境支付：SWIFT gpi、CIPS、区块链
- 风险管理：汇率风险、敞口管理

交易银行

- 现金管理：资金池、流动性管理

- 供应链金融：应收账款融资、区块链确权
- 跨境金融：全球现金管理、贸易融资

债券业务

- 债券发行：簿记建档、信息披露
- 债券交易：银行间市场、交易所市场
- 债券结算：DVP结算、托管服务

工商银行业务

- 对公核心：存款、贷款、支付结算
- 企业网银：现金管理、银企直联
- 投行业务：债券承销、并购重组

共性技术

- 高性能架构：读写分离、分库分表、多级缓存
- 分布式事务：TCC、Saga、消息队列
- 监管合规：RegTech、数据治理

文档版本：v1.0

更新日期：2026-02-08