基于联邦学习新技术连接数据孤岛

陈天健

微众银行 人工智能部 副总经理

Geekbang. 极客邦科技 InfoQ

极客邦科技
会议推荐2019

5月

QCon 北京
全球软件开发大会
大会：5月6–8日
培训：5月9–10日

QCon 广州
全球软件开发大会
培训：5月25–26日
大会：5月27–28日

6月

GTLC GLOBAL TECH LEADERSHIP CONFERENCE 上海
技术领导力峰会
时间：6月14–15日

GMTC 北京
全球大前端技术大会
大会：6月20–21日
培训：6月22–23日

7月

ArchSummit 深圳
全球架构师峰会
大会：7月12–13日
培训：7月14–15日

10月

QCon 上海
全球软件开发大会
大会：10月17–19日
培训：10月20–21日

11月

GMTC 深圳
全球大前端技术大会
大会：11月8–9日
培训：11月10–11日

AiCon 北京
全球人工智能与机器学习大会
大会：11月21–22日
培训：11月23–24日

12月

ArchSummit 北京
全球架构师峰会
大会：12月6–7日
培训：12月8–9日

Geekbang 极客邦科技 | InfoQ 极客传媒

InfoQ官网 全新改版上线

促进软件开发领域知识与创新的传播

关注InfoQ网站
第一时间浏览原创IT新闻资讯

免费下载迷你书
阅读一线开发者的技术干货

About The SPEAKER

陈天健　专家工程师

- 现任微众银行人工智能部副总经理
- 前百度金融首席架构师
- 前百度主任架构师 T10

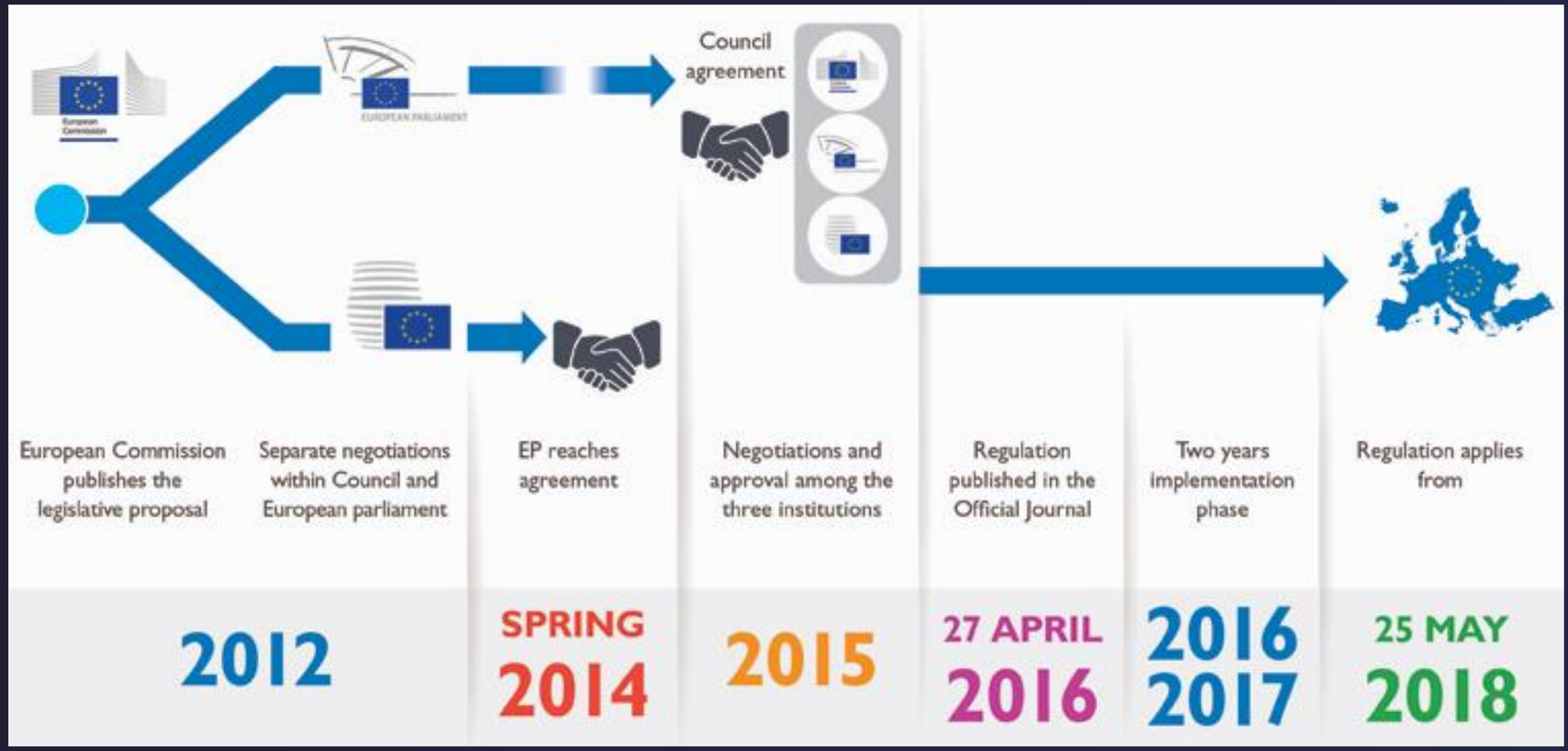
TABLE OF

CONTENTS 大 纲

- 什么是联邦学习
- 联邦学习的工业实践
- 联邦学习生态及相关开源项目

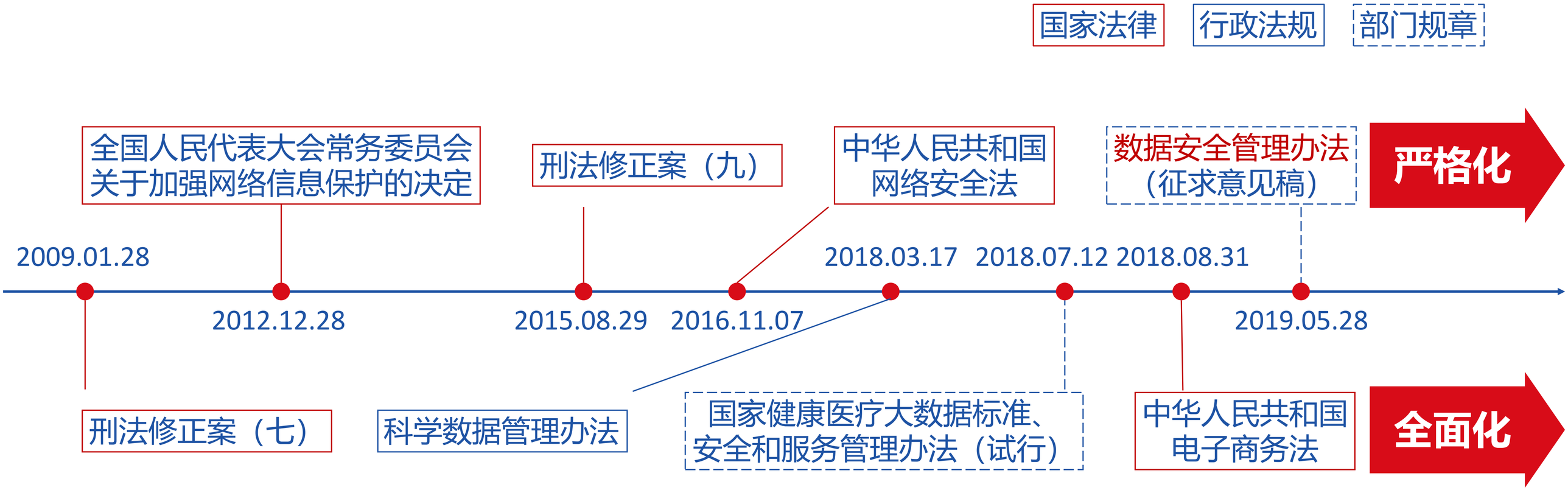
什么是联邦学习？

GDPR Timeline



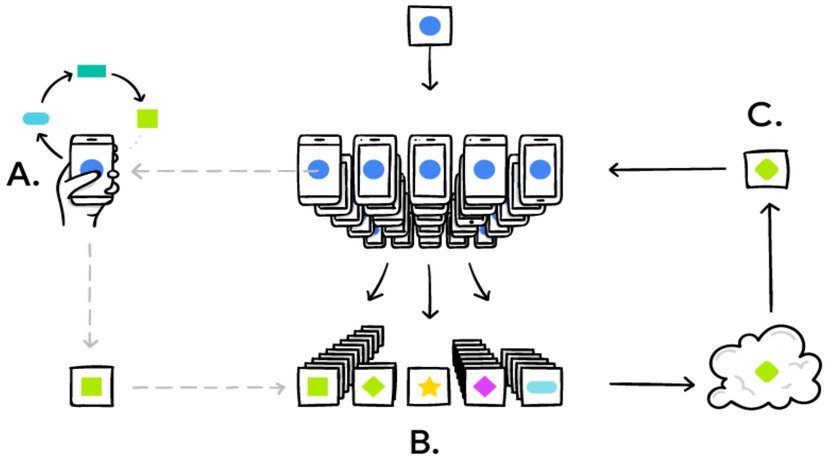
Src: https://www.dlapiper.com/en/norway/focus/eu-data-protection-regulation/background/

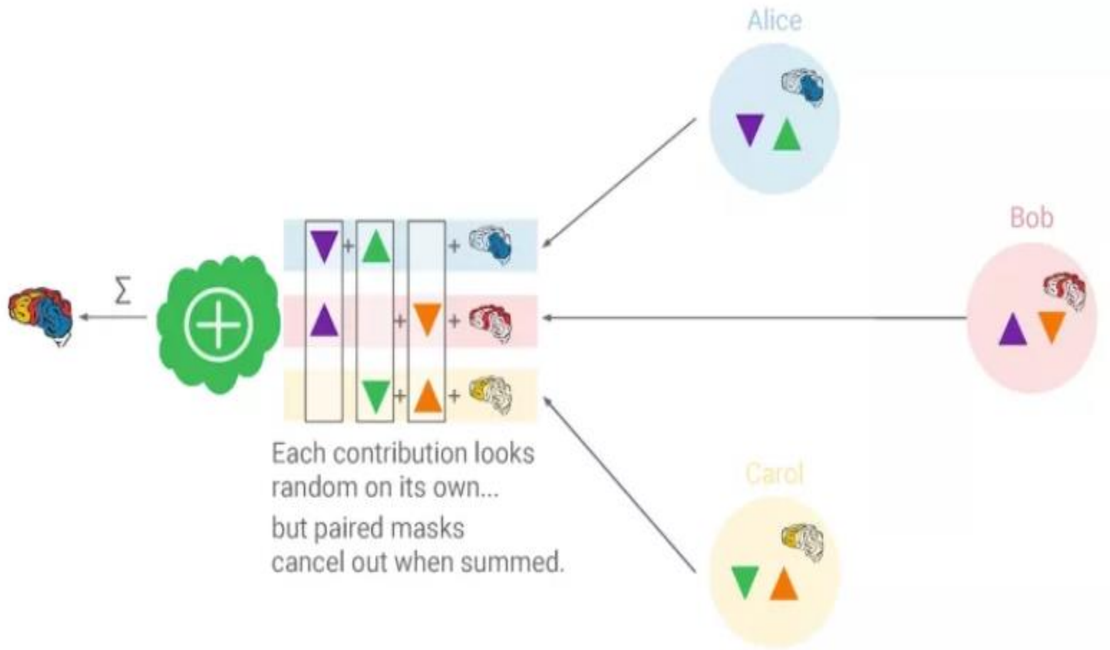
国内数据监管法律体系研究



严格化：数据控制方责任明确，刑罚到自然人
全面化：各领域数据管理细则密集出台，用户授权+监管部门审批

2017 年 Google 发表 Federated Learning





H. Brendan McMahan et al

Communication-Efficient Learning of Deep Networks from Decentralized Data

Google, 2017

Bonawitz K, Ivanov V, Kreuter B, et al.

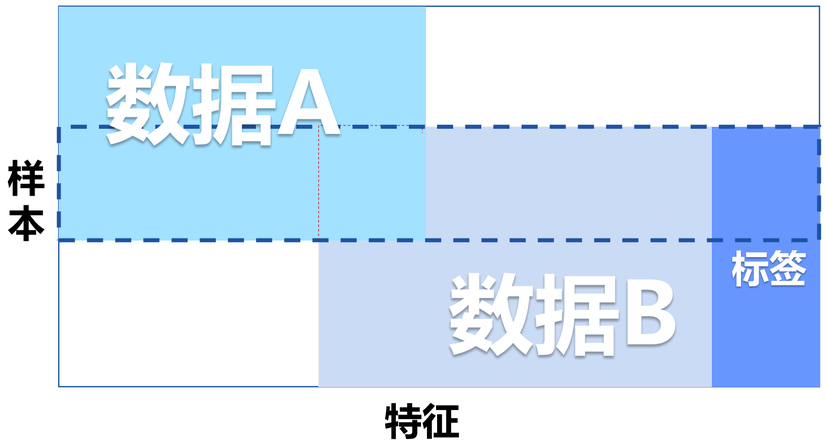
Practical secure aggregation for privacy-preserving machine learning

Google, 2017

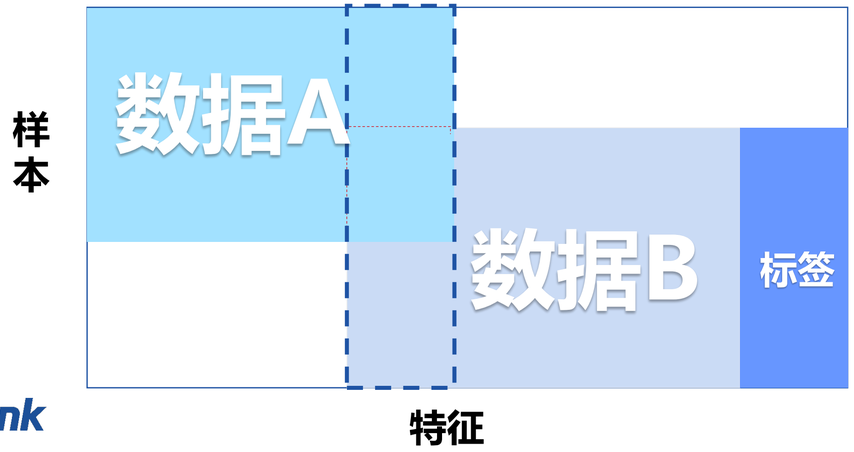
WeBank

完整的联邦学习技术体系

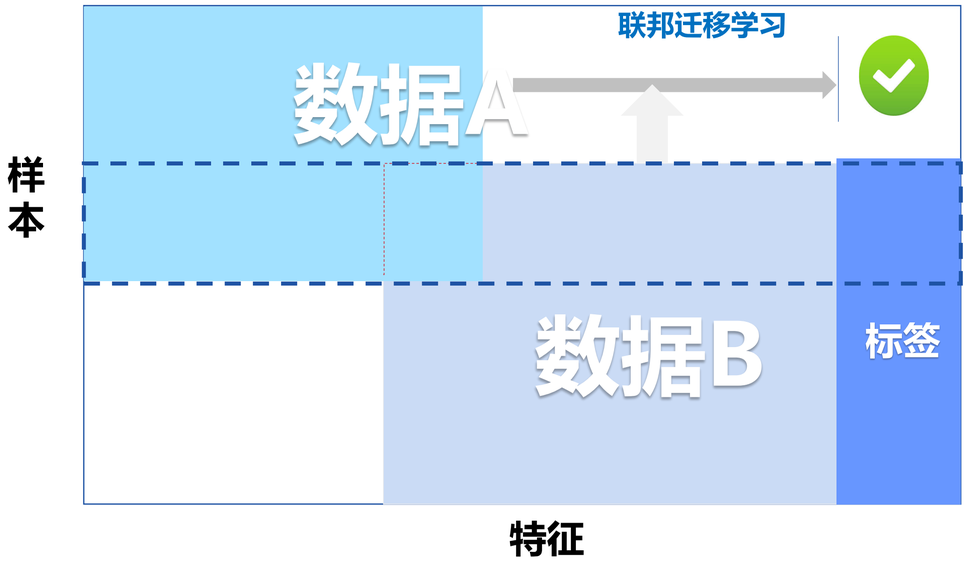
纵向联邦学习



横向联邦学习



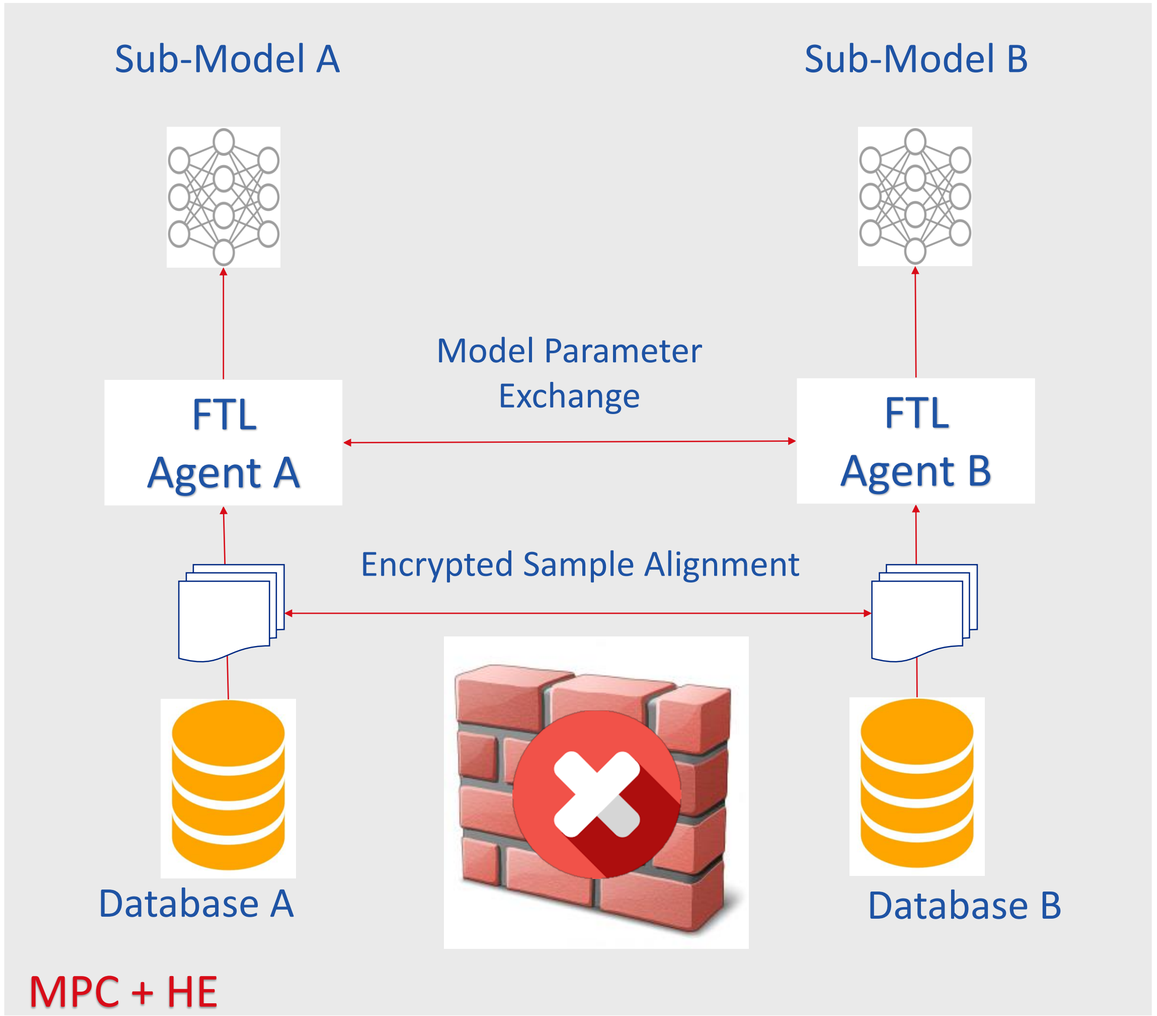
联邦迁移学习



WeBank

联邦学习的工业实践

联邦学习技术加速大数据合作生态构建



银行+监管
联合反洗钱建模

- ✓ 案例召回率提升15%
- ✓ 人工审批效率提升50%

互联网+银行
联合信贷风控建模

- ✓ 数据合作壁垒降低
- ✓ 模型效果提升7%

互联网+保险
联合权益定价建模

- ✓ 定价准确率大幅提升
- ✓ 解决新客覆盖问题
- ✓ 个性化定价覆盖率超90%

互联网+零售
联合客户价值建模

- ✓ 营销效率提升25%
- ✓ 库存去化周期降低

特性	联邦迁移学习	差分隐私	可信沙箱
数据出库	不出库	出库	出库
模型精度	无损	有损	无损
样本容量	无限制	无限制	有限制
开放生态	开放开源	私有项目	第三方运营

Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2018. *Federated Learning*. Communications of The CCF，14, 11 (2018), 49–55

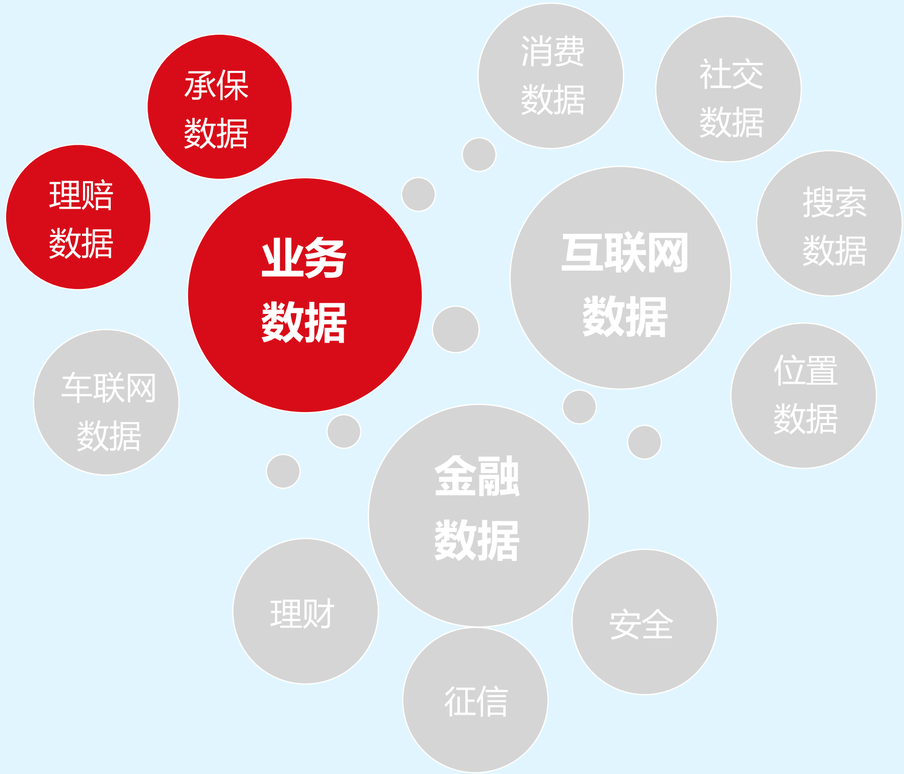
UC1：保险业的个性化定价难题

保险公司理想的数据集

精准个性化用户画像
- 几百维丰富画像
- 多方数据

数据覆盖全面
- ID高匹配
- 全运营商全地域覆盖

保险公司骨感的现实



对客户缺乏全面了解
- 只有10~20维
- 通常只有交易数据

数据分布倾斜严重
- 有重复表现的客户小于10%
- 新客通常没有相关数据

UC1: 基于联邦学习的保险定价

通过对年龄、职业、年租车次数等标签属性进行联邦学习建模，预测出险概率，决策是否出险

互联网企业

ID 客户识别号	X1 年龄	X2 职业
U1	30	金融
U2	30	销售
U3	30	医生
U4	55	金融
U9	50	金融
U10	30	司机
U11	20	学生

1 Billion

人口属性 | 兴趣爱好 | 教育信息 | 财务状况 — 200维度

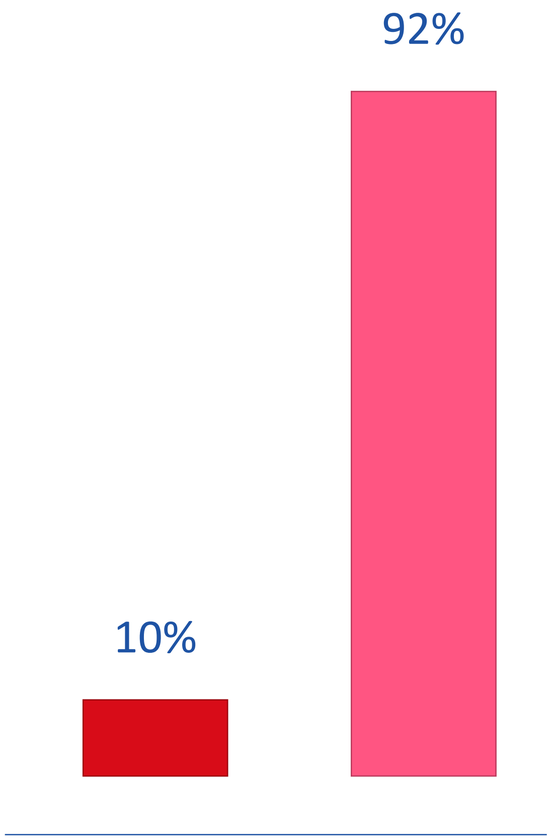
直接承保人

ID 客户识别号	X3 年租车次数	Y 出险概率
U1	1	0.11
U2	5	0.85
U3	2	0.15
U4	1	0.89
U5	1	0.81
U6	5	0.90
U7	1	0.93

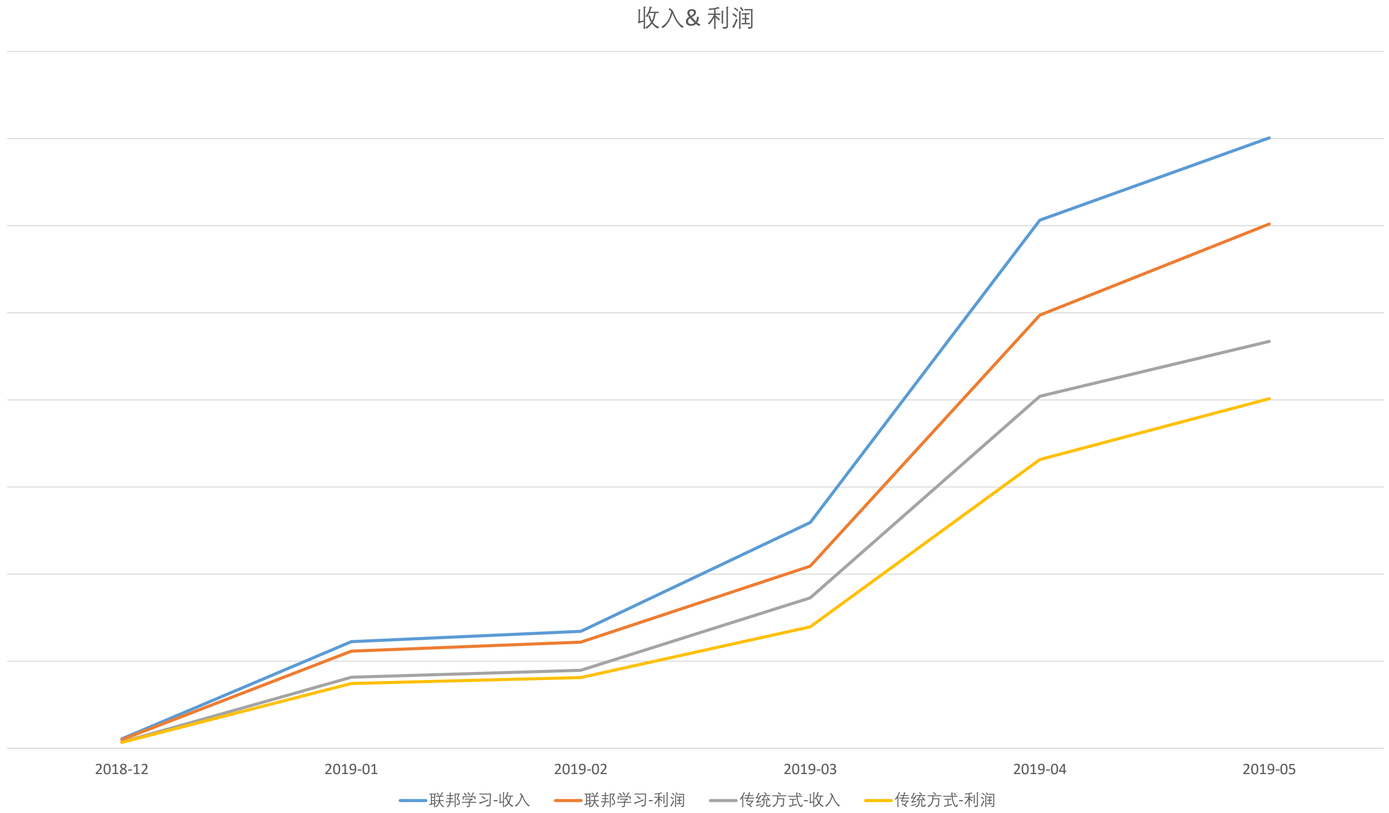
3 Million

30维度 — 车辆信息 | 订单信息 | 业务表现

UC1: 联邦学习解决方案效果



保险权益个性化定价占比提升8倍

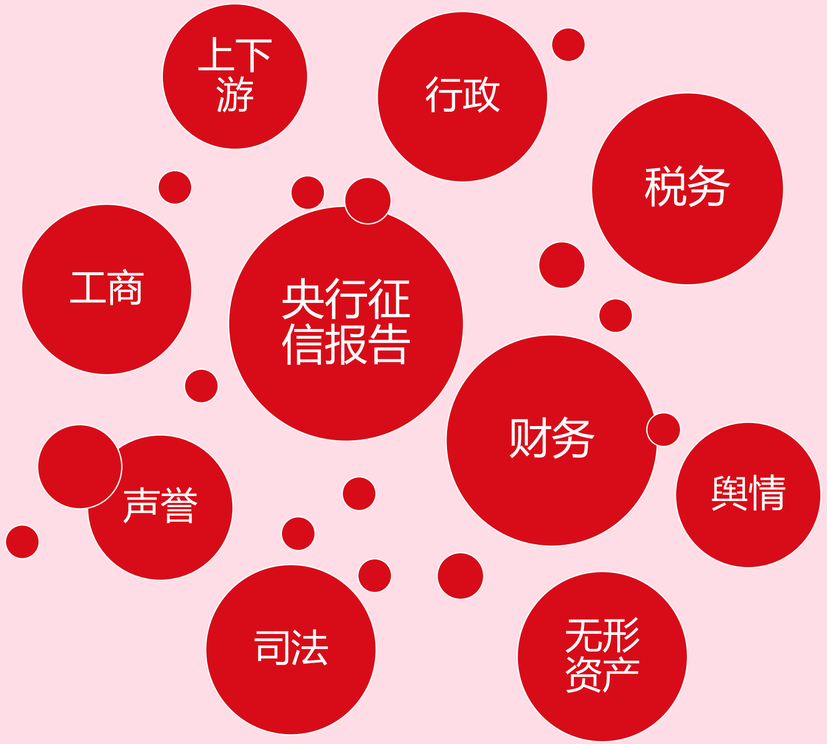


利润提升1.5倍

注: 个性化定价占比 = 个性化定价订单量 / 总体订单量

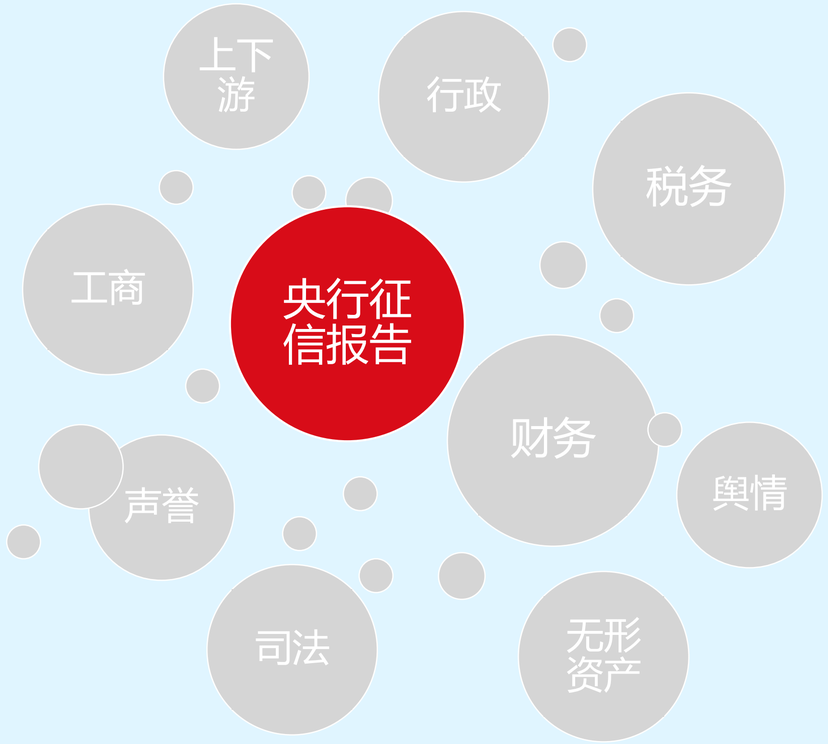
UC2：小微企业信贷的风险管理难题

银行理想的数据集



全面的企业及其控制人相关数据

银行骨感的现实



对客户缺乏全面了解

- 通常只有央行信用报告

数据分布倾斜严重

- 有重复表现的客户小于10%
- 70%客户无任何信用表现

UC2: 基于联邦学习的企业风控模型

通过发票数据、央行征信分等标签属性进行联合建模，预测小微企业信贷逾期概率

合作企业

ID 纳税人识别号	X1 近3个月开票金额	X2 近6个月开票金额
U1	90	150
U2	40	100
U3	20	80
U4	100	200
U9	50	50
U10	6	10
U11	2	5

60 Million

发票数据衍生变量 — 200 维度

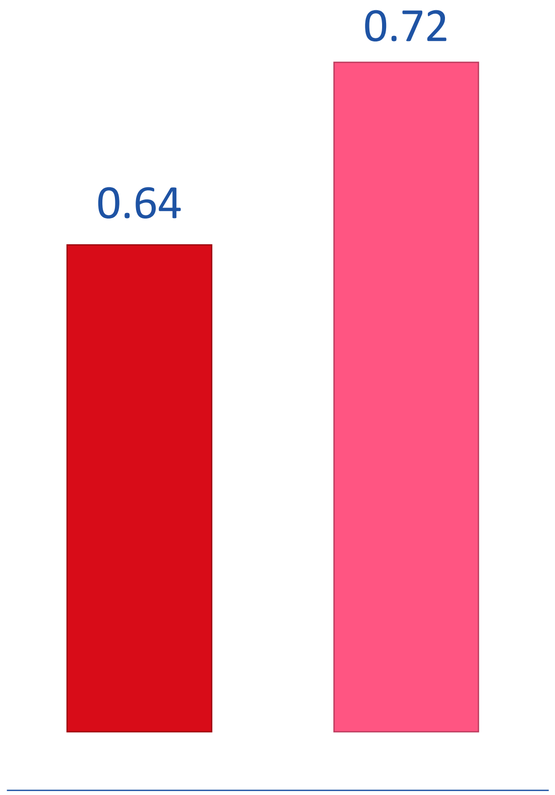
微众银行

ID 纳税人识别号	X3 央行征信分	Y 逾期概率
U1	600	0.79
U2	550	0.11
U3	520	0.88
U4	600	0.15
U5	600	0.90
U6	520	0.81
U7	600	0.32

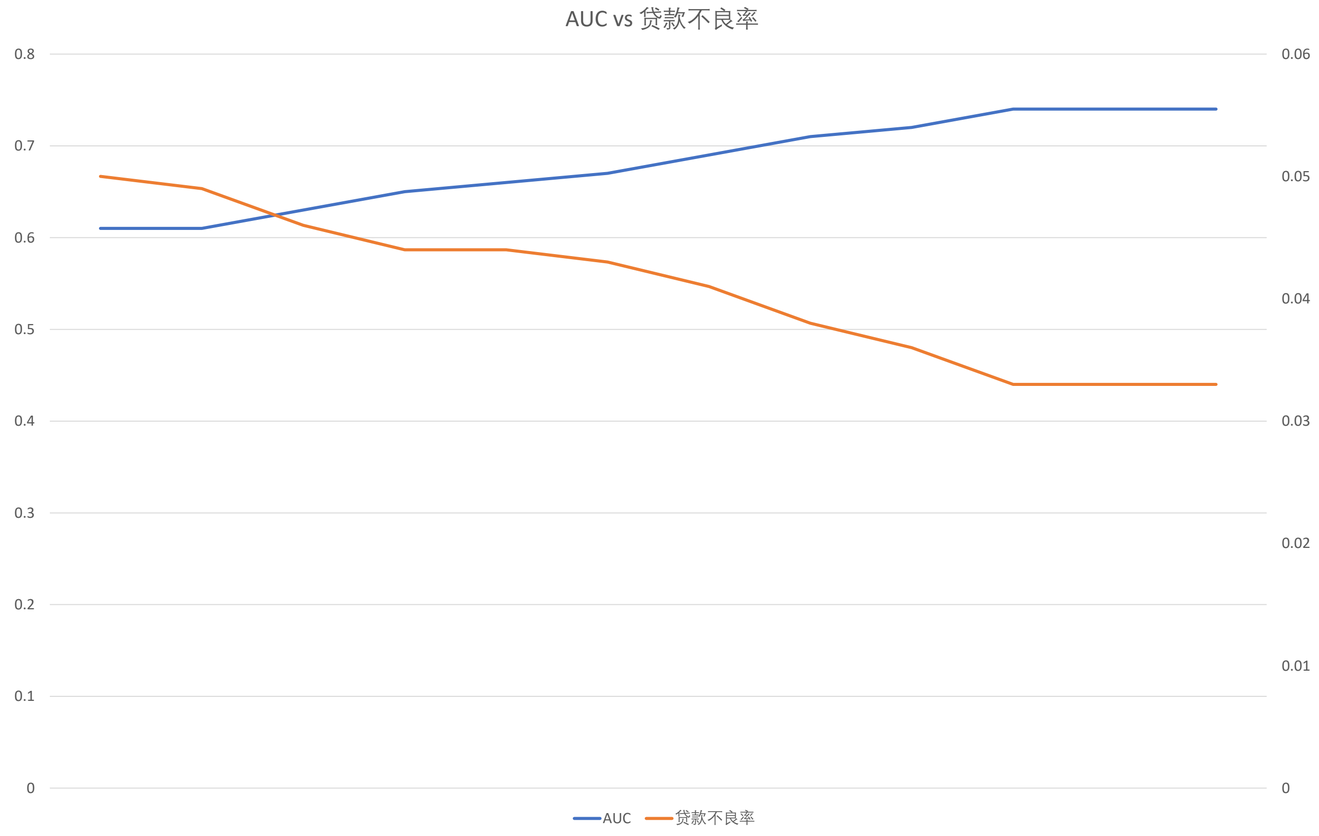
400 K

20 维度 — 税务属性 | 金融属性 | 业务表现

UC2：联邦学习解决方案效果



小微企风控模型区分度提升12%(AUC)



注：AUC是衡量模型区分好坏样本的评估标准之一。AUC越接近0.5，模型预测结果越随机；AUC越接近1.0，模型预测结果越准确。

联邦学习生态及相关开源项目

遇见FATE

愿景

- 工业级别联邦学习系统
- 有效帮助多个机构在符合数据安全和政府法规前提下，进行数据使用和联合建模

设计原则

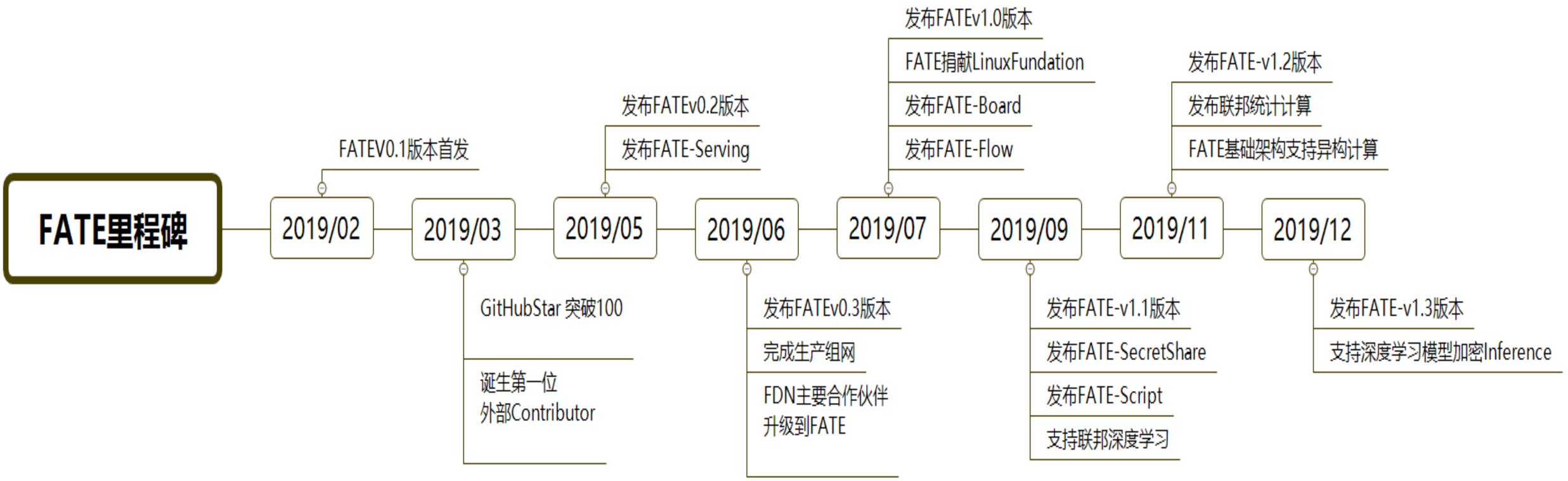
- 支持多种主流算法：为机器学习、深度学习、迁移学习提供高性能联邦学习机制
- 支持多种多方安全计算协议：同态加密、秘密共享、哈希散列等
- 友好的跨域交互信息管理方案，解决了联邦学习信息安全审计难的问题

首次发布

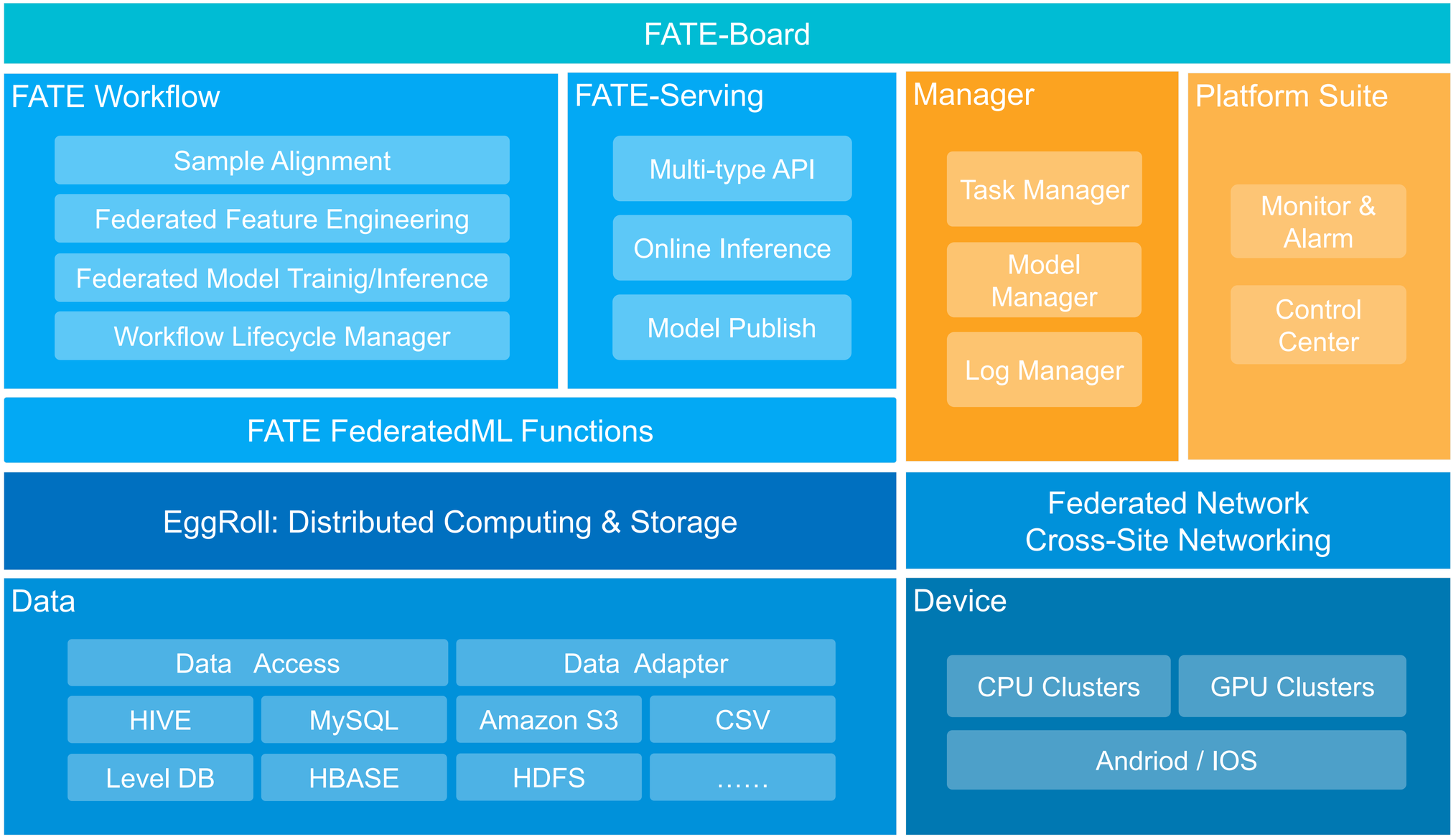
2019年1月份，FATE宣布对外开源

Github：https://github.com/WeBankFinTech/FATE

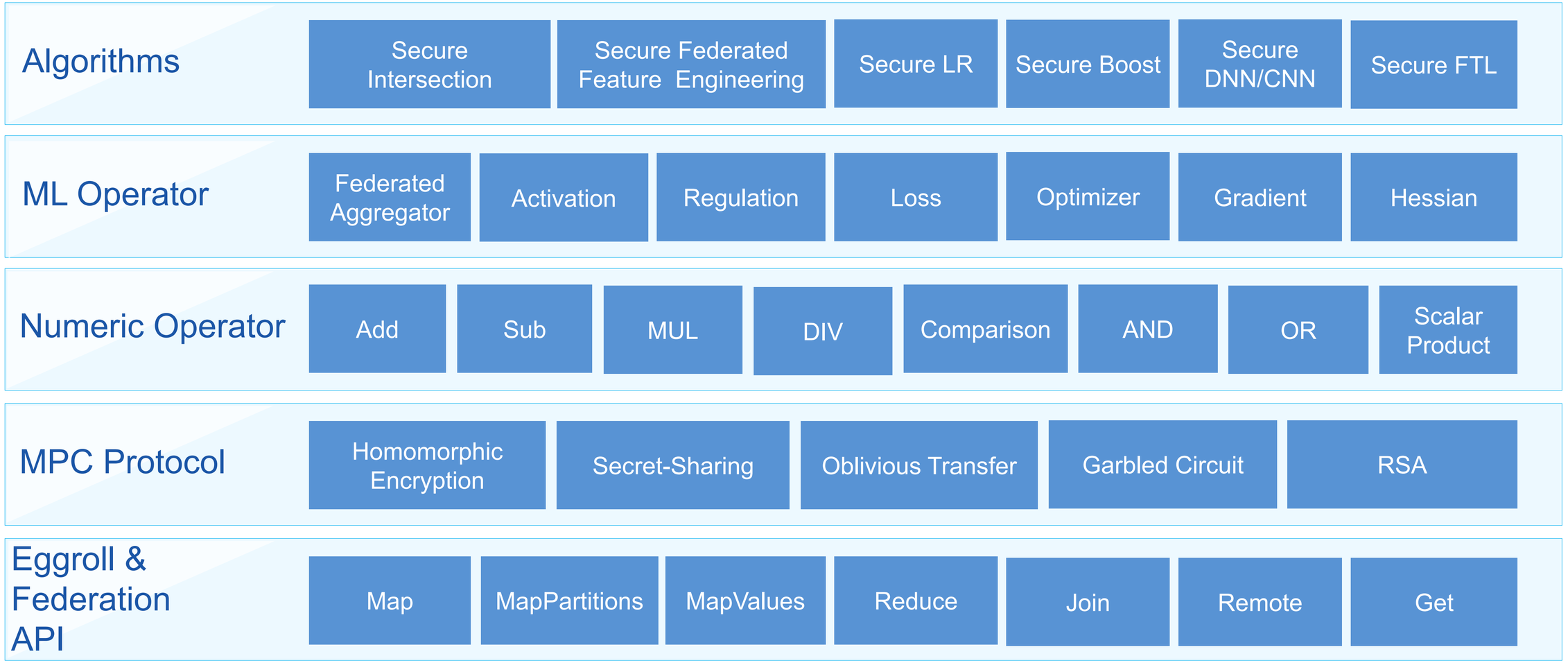
里程碑


FATE里程碑
2019/02
FATEV0.1版本首发
2019/03
GitHubStar 突破100
诞生第一位
外部Contributor
2019/05
发布FATEv0.2版本
发布FATE-Serving
2019/06
发布FATEv0.3版本
完成生产组网
FDN主要合作伙伴
升级到FATE
2019/07
发布FATEv1.0版本
FATE捐献LinuxFundation
发布FATE-Board
发布FATE-Flow
2019/09
发布FATE-v1.1版本
发布FATE-SecretShare
发布FATE-Script
支持联邦深度学习
2019/11
发布FATE-v1.2版本
发布联邦统计计算
FATE基础架构支持异构计算
2019/12
发布FATE-v1.3版本
支持深度学习模型加密Inference

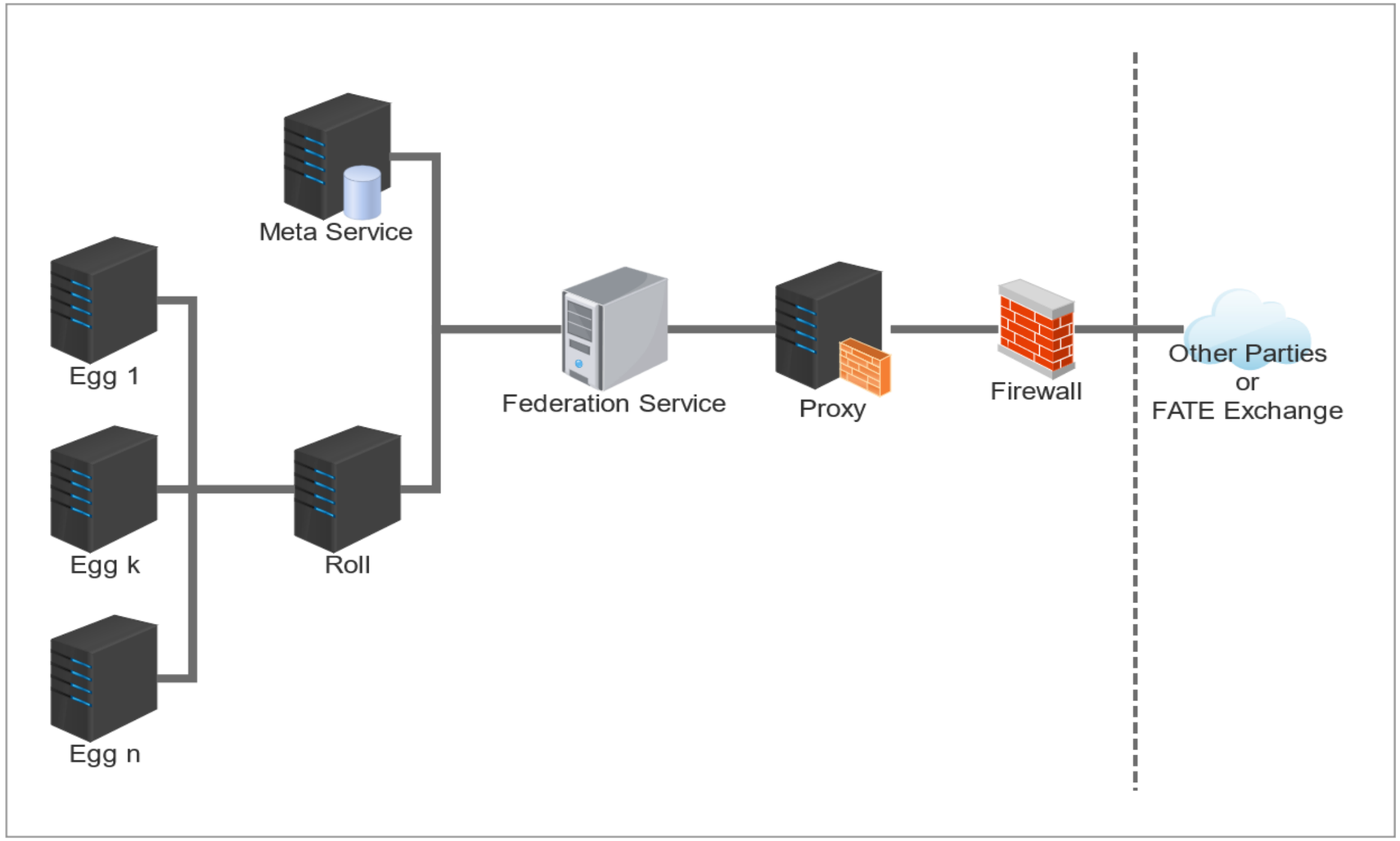
技术架构总览


FATE-Board
FATE Workflow
Sample Alignment
Federated Feature Engineering
Federated Model Trainig/Inference
Workflow Lifecycle Manager
FATE-Serving
Multi-type API
Online Inference
Model Publish
Manager
Task Manager
Model Manager
Log Manager
Platform Suite
Monitor & Alarm
Control Center
FATE FederatedML Functions
EggRoll: Distributed Computing & Storage
Federated Network
Cross-Site Networking
Data
Data Access
Data Adapter
HIVE
MySQL
Amazon S3
CSV
Level DB
HBASE
HDFS
……
Device
CPU Clusters
GPU Clusters
Andriod / IOS

FATE FederatedML Functions



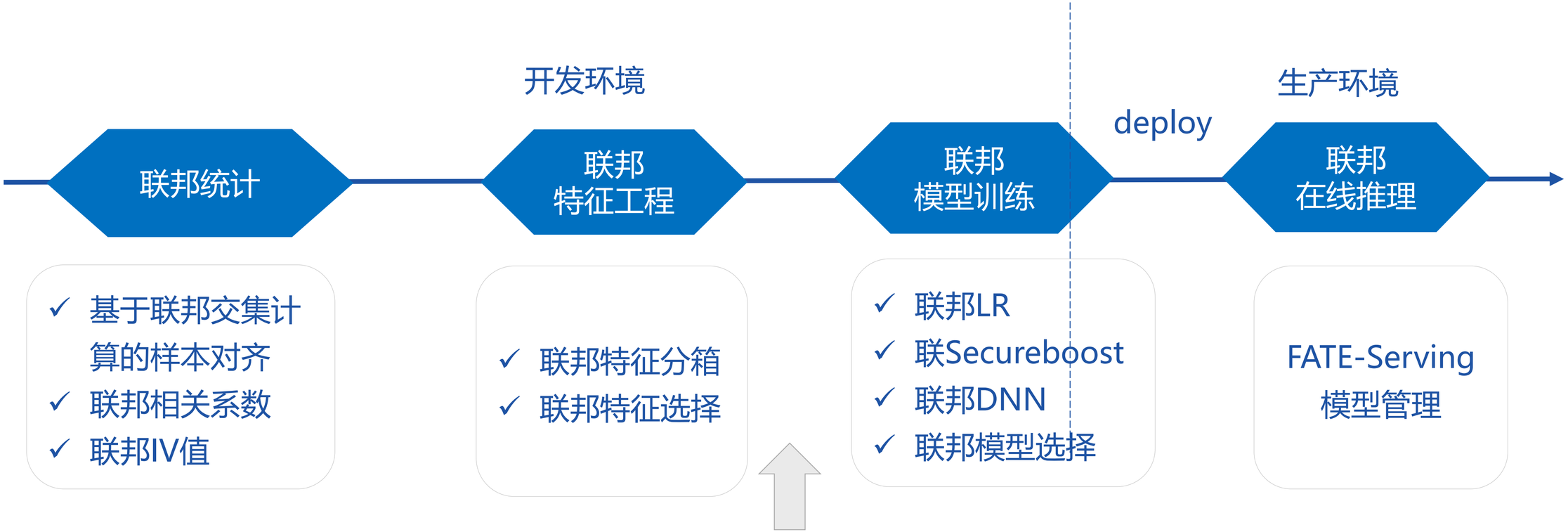
一方部署网络拓扑-示例


Meta Service
Egg 1
Egg k
Egg n
Roll
Federation Service
Proxy
Firewall
Other Parties
or
FATE Exchange

目前 FATE 项目中算法&案例

- Secure Intersection for Sample Alignment
- Vertical-Split Feature Space Federated Feature Engineering
  - Secure Feature Binning
  - Secure Feature Selection
  - Secure Feature Correlation（Coming Soon）
- Vertical-Split Feature Space Federated Learning
  - Secure Logistic Regression
  - Secure Boosting Tree
  - Secure DNN/CNN（Coming Soon）
- Horizontal-Split Sample Space Federated Learning
  - Secure Logistic Regression
  - Secure Boosting Tree（Coming Soon）
  - Secure DNN/CNN（Coming Soon）
- Secure Federated Transfer Learning

一站式联合建模Pipeline



如果想开发新的联邦学习算法呢?

开发流程

1 → 2 → 3 → 4

选择一个机器学习算法，设计多方安全计算协议

定义多方交互的数据变量

构建算法执行工作流

基于EggRoll & Federation Api 实现算法工作流中各个功能组件

WorkFlow Example

- 工作流
  - 定义联邦算法组件执行工作流
  - 组件
    - 参数初始化组件，
    - 数据加载和转换组件
    - 训练、预测组件
    - 评估组件
    - 模型保存组件
    - .....

```
def run(self):
    self._init_argument()
    if self.workflow_param.method == "train":
        train_data_instance = None
        predict_data_instance = None
        if self.role != consts.ARBITER:
            train_data_instance = self.gen_data_instance(self.workflow_param.train_input_table,
                                                         self.workflow_param.train_input_namespace)
            if self.workflow_param.predict_input_table is not None and self.workflow_param.predict_input_namespace
                predict_data_instance = self.gen_data_instance(self.workflow_param.predict_input_table,
                                                               self.workflow_param.predict_input_namespace)
        self.train(train_data_instance, validation_data=predict_data_instance)
```

```
def train(self, train_data, validation_data=None):
    self.model.fit(train_data)
    self.save_model()

    if self.role == consts.GUEST or self.role == consts.HOST or \
            self.mode == consts.HOMO:
        eval_result = {}
        predict_result = self.model.predict(train_data,
                                            self.workflow_param.predict_param)
        train_eval = self.evaluate(predict_result)
        eval_result[consts.TRAIN_EVALUATE] = train_eval
        if validation_data is not None:
            val_pred = self.model.predict(validation_data,
                                          self.workflow_param.predict_param)
            val_eval = self.evaluate(val_pred)
            eval_result[consts.VALIDATE_EVALUATE] = val_eval
        self.save_eval_result(eval_result)
```

FederatedML Functions Example

- **纵向LR梯度一方分布式计算**
  - 定义梯度和损失计算公式
  - 设计算法并行方式
  - 通过Eggroll API 实现分布式梯度聚合和损失计算

```python
class HeteroLogisticGradient(object):
    .........

    def compute_fore_gradient(self, data_instance, encrypted_wx):
        fore_gradient = encrypted_wx.join(data_instance, lambda wx, d: 0.25 * wx - 0.5 * d.label)
        return fore_gradient

    def compute_gradient(self, data_instance, fore_gradient, fit_intercept):
        feat_join_grad = data_instance.join(fore_gradient, lambda d, g: (d.features, g))
        f = functools.partial(self.__compute_gradient, fit_intercept=fit_intercept)
        gradient_partition = feat_join_grad.mapPartitions(f)
        gradient = HeteroFederatedAggregator.aggregate_mean(gradient_partition)
        return gradient

    def compute_gradient_and_loss(self, data_instance, fore_gradient,
                                  encrypted_wx, en_sum_wx_square, fit_intercept):
        # compute gradient
        gradient = self.compute_gradient(data_instance, fore_gradient, fit_intercept)

        # compute and loss
        half_ywx = encrypted_wx.join(data_instance, lambda wx, d: 0.5 * wx * int(d.label))
        half_ywx_join_en_sum_wx_square = half_ywx.join(en_sum_wx_square, lambda yz, ez: (yz, ez))
        f = functools.partial(self.__compute_loss)
        loss_partition = half_ywx_join_en_sum_wx_square.mapPartitions(f)
        loss = HeteroFederatedAggregator.aggregate_mean(loss_partition)

        return gradient, loss
```

Federation API Example

- **纵向LR梯度两方联合**
  - 定义算法交互信息-梯度（json 配置文件，数据源和目的地）
  - 生成梯度交互信息唯一标识符
  - Federation API 完成梯度交互信息的收发

```
"HeteroLRTransferVariable": {
  "host_forward_dict": {
    "src": "host",
    "dst": [
      "guest"
    ]
  },
  "fore_gradient": {
    "src": "guest",
    "dst": [
      "host"
    ]
  },
  "guest_gradient": {
    "src": "guest",
    "dst": [
      "arbiter"
    ]
  },
  "guest_optim_gradient": {
    "src": "arbiter",
    "dst": [
      "guest"
    ]
  },
  "host_loss_regular": {
    "src": "host",
    "dst": [
      "guest"
    ]
  },
```

```
self.transfer_variable = HeteroLRTransferVariable()

federation.remote(guest_gradient,
                  name=self.transfer_variable.guest_gradient.name,
                  tag=self.transfer_variable.generate_transferid(
                      self.transfer_variable.guest_gradient,
                      self.n_iter_,
                      batch_index),
                  role=consts.ARBITER,
                  idx=0)
```

```
optim_guest_gradient = federation.get(name=self.transfer_variable.guest_optim_gradient.name,
                                      tag=self.transfer_variable.generate_transferid(
                                          self.transfer_variable.guest_optim_gradient, self.n_iter_,
                                          batch_index),
                                      idx=0)
```

更多资源请访问FedAI官网

https://FedAI.org/

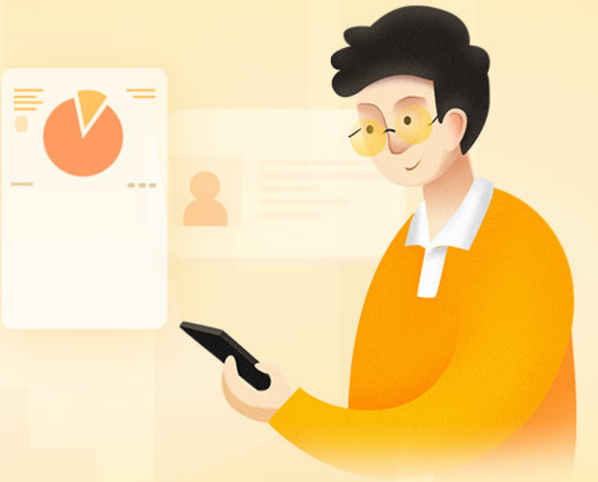
WeChat

极客时间全部课程任学

喊老板来买单！

- 精选 13+ 热门职位的学习路径，包括架构、运维、前端工程师等
- 根据不同技术岗位能力模型匹配合适的课程
- 一键设置购买条件，成员按需选课，自主制定学习计划
- 享充值满赠优惠，帮老板省钱，团队免费学习

立即申请



THANKS

Geekbang. 极客邦科技 InfoQ