



FATE：新一代联邦学习技术及应用实战

范涛

微众银行高级研究员

dylanfan@webank.com



<https://www.fedai.org/>

<https://github.com/WeBankFinTech/FATE>

目录

CONTENTS

1

联邦学习背景介绍

2

纵向联邦学习

3

横向联邦学习

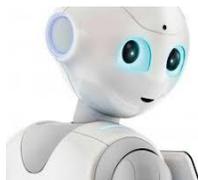
4

联邦学习开源平台-FATE

01

联邦学习背景介绍

AI落地：理想vs现实



理想

数据质量好

标签数据充足

数据集中

现实

数据质量差

缺乏标签数据

数据分散隔离

Twitter taught Microsoft's AI chatbot to be a racist asshole in less than a day

“昔日的人工智能老大哥，IBM WATSON为什么现在会被看作一个笑话？”

80% 以上的企业存在数据孤岛问题 (information silos)



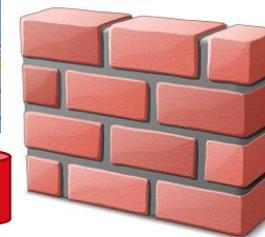
问题:

- 无法了解基因与疾病的关系 (只有医生可以给数据打标签)
- 买数据? Verily Life Sciences 有一万名志愿者, 但需要10年!



企业 A

X1



企业 B

(X2, Y)

基于联邦学习的技术生态

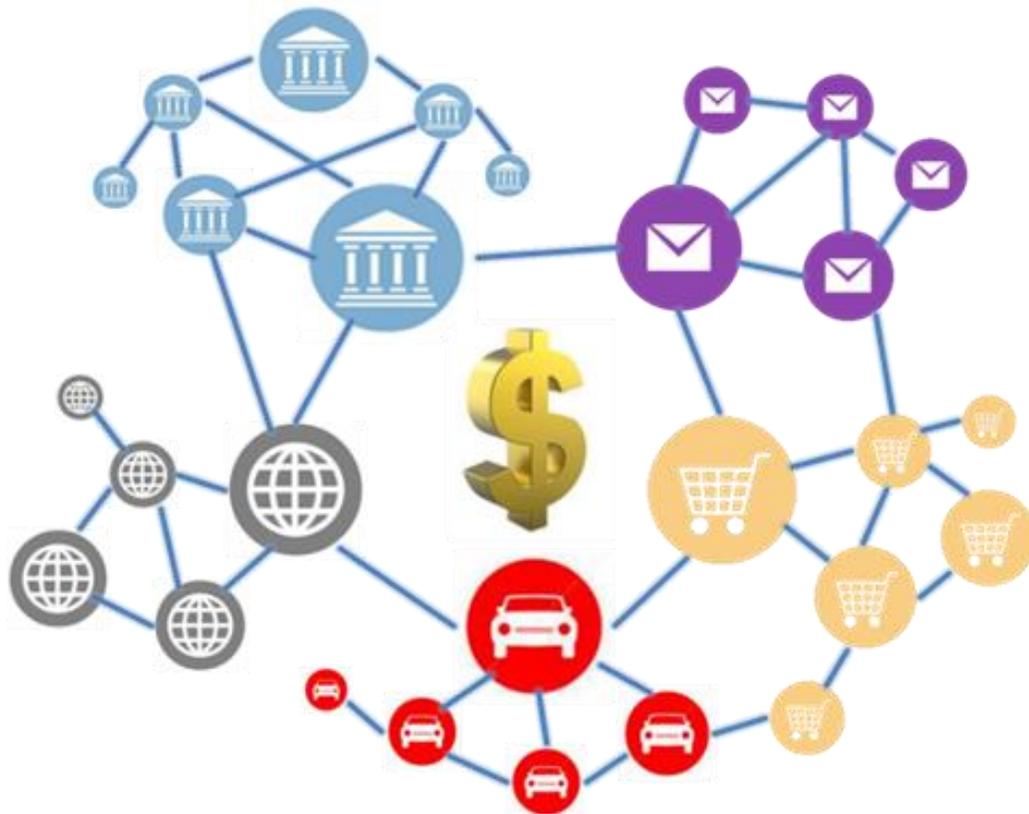
01 数据隔离
数据不泄露到外部

03 对等
参与者地位的对等



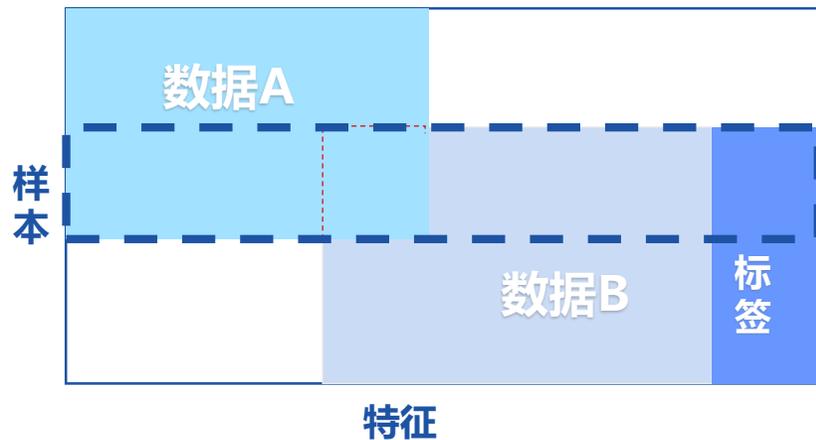
无损 02
联邦模型效率等同或
接近全量数据模型

共同获益 04
参与者共同获益

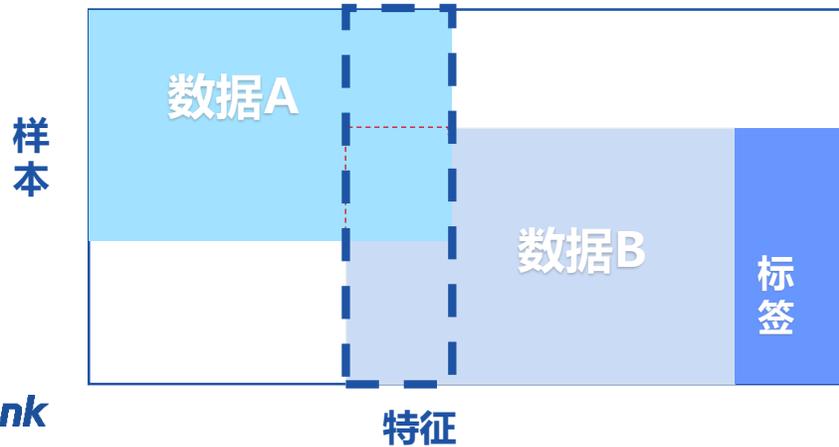


联邦学习的分类体系

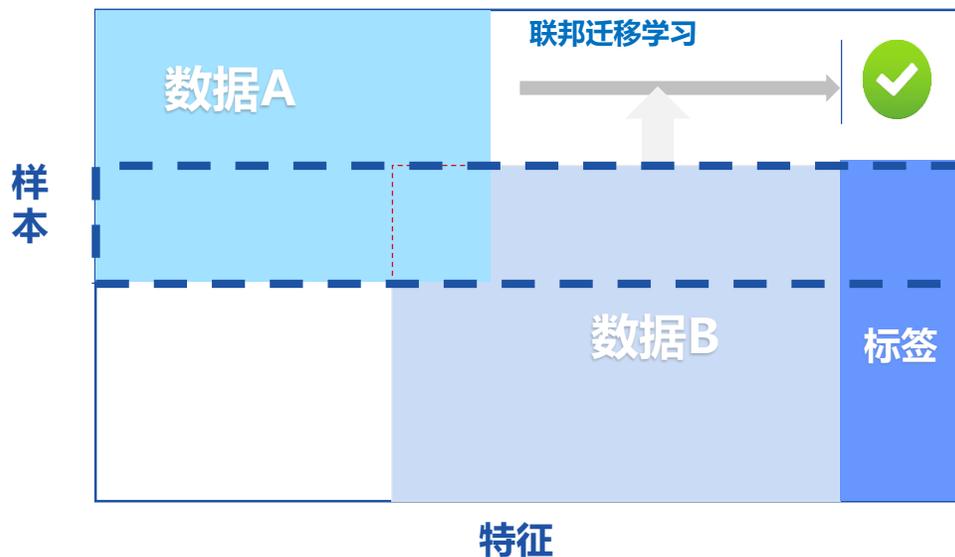
纵向联邦学习



横向联邦学习



联邦迁移学习



02

纵向联邦学习

纵向联邦学习-联合建模需求场景

举例：微众与合作企业联合建模，微众有Y（业务表现），期望优化本方的Y预测模型

◆ 设定：

- ✓ 只有微众拥有 Y= “逾期表现”
- ✓ 合作企业无法暴露含有隐私的 X

◆ 传统建模方法问题：

- ✓ 合作企业缺乏Y无法独立建模
- ✓ X数据全量传输到微众不可行

◆ 期望结果：

- ✓ 保护隐私条件下，建立联合模型
- ✓ 联合模型效果超过单边数据建模

合作企业

ID 证件号 电话号	X1 帐龄	X2 月薪	X3 等级
U1	9	8000	A
U2	4	5000	C
U3	2	3500	C
U4	10	10000	A
U5	5	7500	B
U6	5	7500	A
U7	8	8000	B

业务系统A 数据

微众银行

ID 证件号 电话号	X4 央行征信分	X5 微众内部分	Y 表现数据
U1	600	600	无
U2	550	500	有
U3	520	500	有
U4	600	600	无
U8	600	600	无
U9	520	500	有
U10	600	600	无

业务系统B 数据

同态加密技术保护隐私

◆ 数据隐私保护:

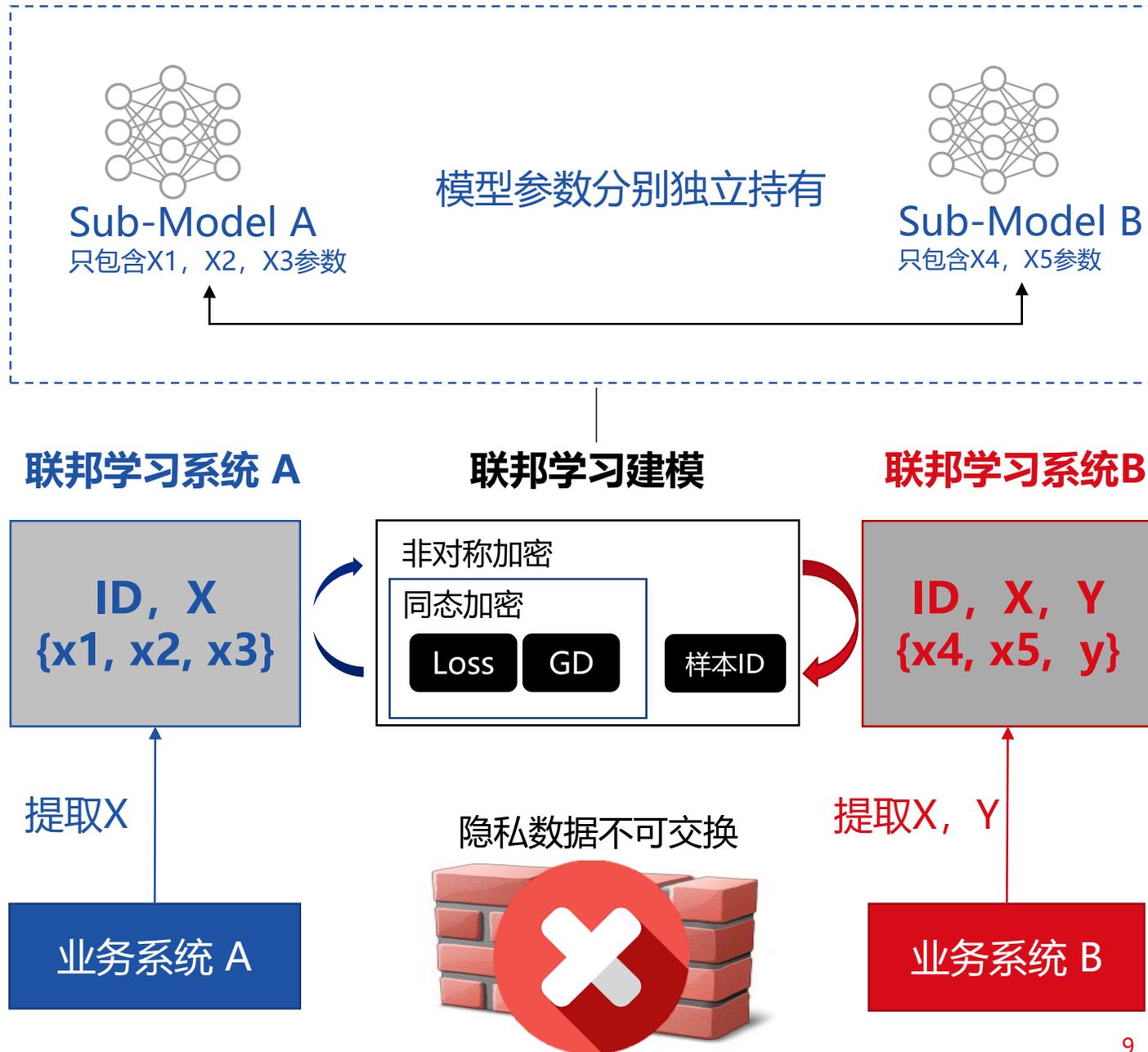
- ✓ 建模样本ID差集不向对方泄露
- ✓ 任何底层X, Y数据不向对方泄露

◆ 模型参数保护:

- ✓ 分别持有, 联合使用

◆ 结果:

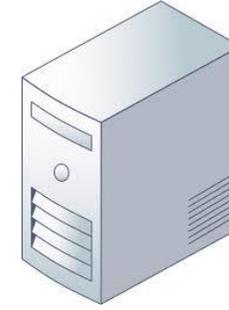
- ✓ A方有A模型
- ✓ B方有B模型
- ✓ A和B模型都比单独建模好



基于隐私保护的样本id匹配



How to find $X \cap Y = [u1, u2, u3]$?
Party A does not know Party B has u5
Party B does not know Party A has u4



[u1, u2, u3, u4]

ID set X
Party A



[u1, u2, u3, u5]

ID set Y
Party B

基于隐私保护的样本id匹配

Party A



$X_A: \{u1, u2, u3, u4\}$

$H(x): x$ 的哈希

public key: (n, e)

Party B



$X_B: \{u1, u2, u3, u5\}$

RSA: n, e, d

$$Y_A = \{ri^{e*}H(ui) \mid ui \in X_A \text{ } ri:rand\}$$

$$D_A = \{H(ri*(H(ui))^d / ri) = H((H(ui))^d) \mid ri*(H(ui))^d \in Z_A\}$$

$$I = D_A \cap Z_B = \{H((H(u1))^d), H((H(u2))^d), H((H(u3))^d)\}$$

$$I, D_A \Rightarrow \{u1, u2, u3\}$$

$$Y_A = \{r1^e H(u1), r2^e H(u2), r3^e H(u3), r4^e H(u4)\}$$

Z_A, Z_B

|

$$Z_A = \{(ri^e H(ui))^d = ri*(H(ui))^d \% n \mid ri^e H(ui) \in Y_A\}$$

$$Z_B = \{H((H(u))^d) \mid u \in X_B\}$$

$$I, Z_B \Rightarrow \{u1, u2, u3\}$$

同态加密

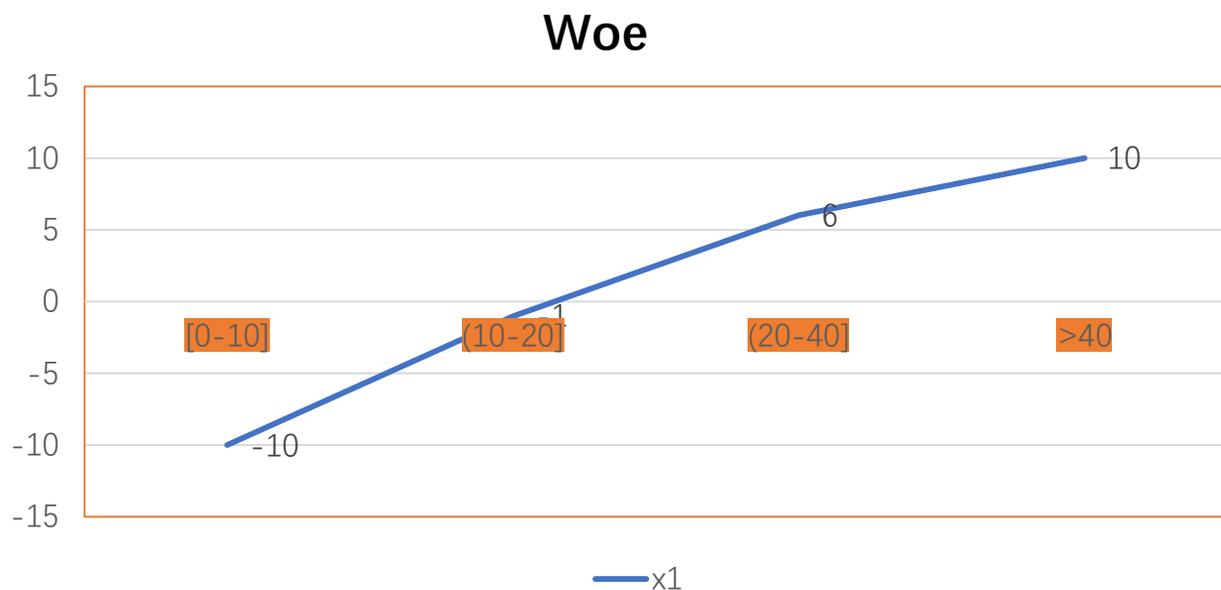
- 全同态或者半同态 Full Homomorphic Encryption and Partial Homomorphic Encryption
- 数据层面的信息保护 Data-level information protection

Paillier 半同态加密 Partially homomorphic encryption

$$\begin{aligned} \text{Addition :} & \quad [[u]] + [[v]] = [[u+v]] \\ \text{Scalar multiplication:} & \quad n[[u]] = [[nu]] \end{aligned}$$

Rivest, R. L.; Adleman, L.; and Dertouzos, M. L. 1978. On data banks and privacy homomorphisms. Foundations of Secure Computation, Academia Press, 169–179.

联邦特征工程



特征IV值

特征	IV
x1	0.3
x2	0.2
x3	0.4
x4	0.01
x5	0.05

问题：在保护双方隐私下，A侧(含X)和B侧(含X, Y)特征如何计算WOE和IV?

难点：

- A 侧只有特征x，没有y；计算Woe和IV得同时依赖x,y (B侧特征Woe &IV 可以本地计算)
- A侧不能对B侧暴露x，B侧不能对A侧暴露y
- 最终只能让B侧获得所有特征Woe & IV

A: X_i (特征分箱)

id_set_1 (eg: id1, id2, id5,..)
id_set_2

id_set_i

id_set_k

Encry(x): x的加法同态加密,
Encode(x): 本地编码

1. {idi , Encry(yi), Encry(1-yi)}

2. {
Encode(id_set_i),
sum(Encry(yi)),
sum(Encry(1-yi))
}

B: (X,Y)

id1	y1
id2	y2
***	***
idi	yi
***	****
idn	yn

3.
npos_i =
Decry(sum(Encry(yi)));
nneg_i =
Decry(sum(Encry(1-yi)))

B 方本地计算

1. $distpos_i = npos_i / pos_total$; $distneg_i = nneg_i / neg_total$
2. $Woe_i = 100 * \log(distpos_i / distneg_i)$
3. $IV = \sum_{i=1}^k (dispos_i - disneg_i) * \log(dispos_i / disneg_i)$

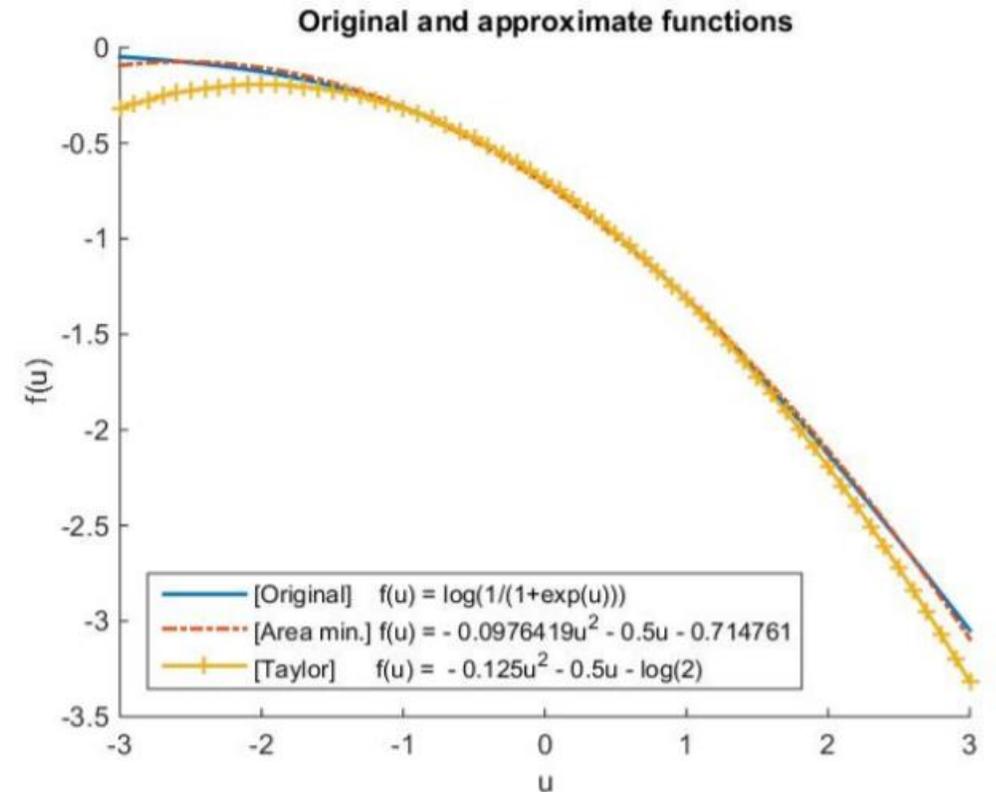
同态加密在机器学习上应用

多项式近似 Polynomial approximation for logarithm function

$$\begin{aligned}l(w) &= \log(1 + \exp(-yw^T x)) \\ &\approx \log 2 - \frac{1}{2} yw^T x + \frac{1}{8} (w^T x)^2 \\ \nabla l(w) &= \left(\frac{1}{1 + \exp(-yw^T x)} - 1 \right) yx \\ &\approx \left(\frac{1}{2} yw^T x - 1 \right) \frac{1}{2} yx\end{aligned}$$

加密计算 Encrypted computation for each term in the polynomial function

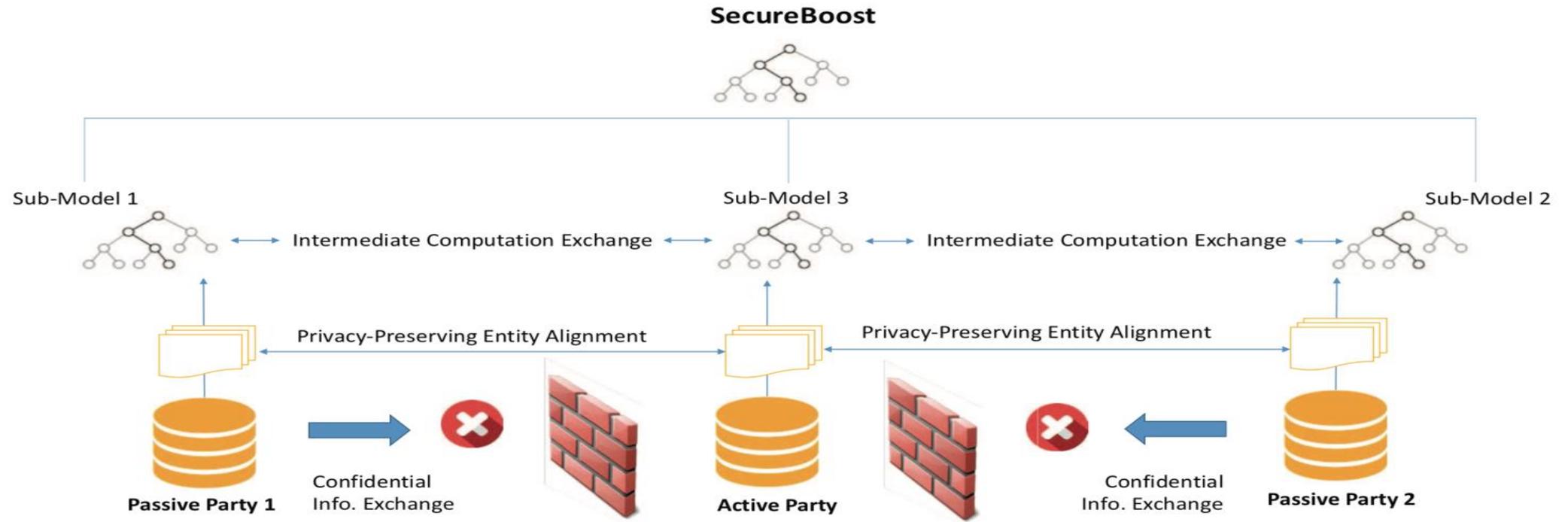
$$\begin{aligned}loss &= \log 2 - \frac{1}{2} yw^T x + \frac{1}{8} (w^T x)^2 \\ [[loss]] &= [[\log 2]] + \left(-\frac{1}{2}\right) * [[yw^T x]] + \frac{1}{8} [[(w^T x)^2]]\end{aligned}$$



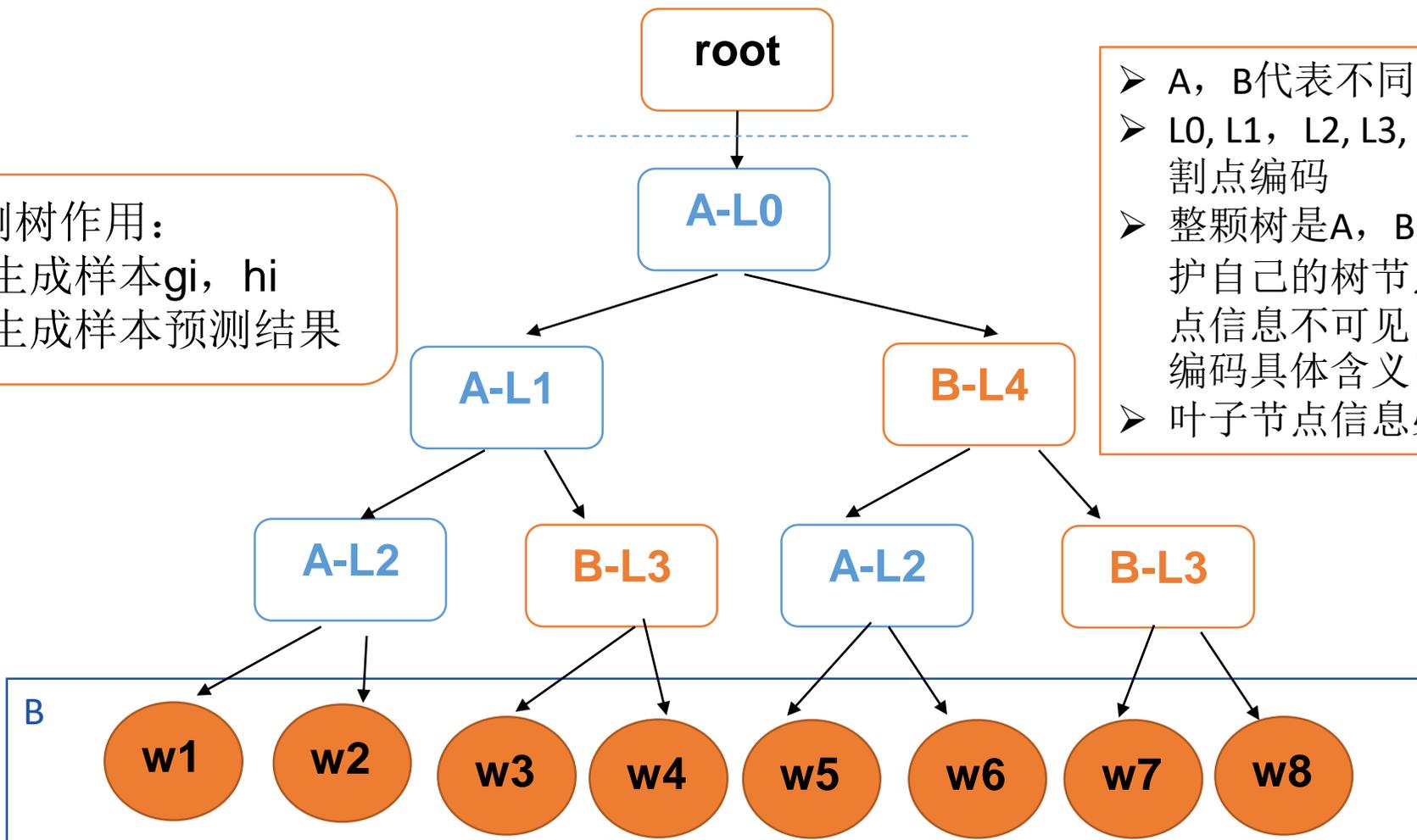
- Kim, M.; Song, Y.; Wang, S.; Xia, Y.; and Jiang, X. 2018. Secure logistic regression based on homomorphic encryption: Design and evaluation. JMIR Med Inform 6(2)
- Y. Aono, T. Hayashi, T. P. Le, L. Wang, Scalable and secure logistic regression via homomorphic encryption, CODASPY16

SecureBoost

- Collaboratively learn a shared gradient-tree boosting model
- Lossless meanwhile secure



Xgboost预测树作用：
训练阶段：生成样本 g_i, h_i
预测阶段：生成样本预测结果



- A, B代表不同数据owner
- L0, L1, L2, L3, L4 代表不同feature的分割点编码
- 整颗树是A, B共同维护, 每一方只维护自己的树节点, 对另外一方的树节点信息不可见 (只知道编码, 不知道编码具体含义)
- 叶子节点信息必须存在B

A: X_i (特征bin)

id_set_1
id_set_2

id_set_i

id_set_k

Step1: $\{id_i, Encry(g_i), Encry(h_i)\}$

Step2: $\{index(id_set_i), sum(Encry(g_i)), sum(Encry(h_i))\}$

Step3: $\{max(gain), argmax(gain)\}$

B

id1	g1	h1
id2	g2	h2
***	***	***
idi	gi	hi
***	****	***
idn	gn	hn

Decry: $sum(Encry(g_i)), sum(Encry(h_i))$
计算 $max(gain), argmax(gain)$

隐私保护:

- 信息增益是在B侧本地计算，B侧没有样本信息泄露
- A本地计算加密后的梯度直方图，B解密梯度直方图，但是不知道具体对应的id集合，保护了A侧id集合隐私信息

03

横向联邦学习

横向联邦学习-联合建模需求场景

举例：微众和合作行共建反洗钱模型，期望优化反洗钱模型

◆ 设定：

- ✓ Y 表示 “是否存在洗钱行为”
- ✓ 合作行和微众都有 (X,Y)
- ✓ 双方不暴露自己的 (X,Y)

◆ 传统建模方法问题：

- ✓ 微众和合作行各自样本不够多

◆ 期望结果：

- ✓ 保护隐私条件下，建立联合模型
- ✓ 联合模型效果超过单边数据建模

微众银行

ID 证件号 电话号	X1 资金来源和 经营范围不符 笔数	X2 大额交易 笔数	Y 表现数据
U1	5	15	有
U2	8	20	有
U3	0	5	无
U4	0	0	无
U5	2	1	无
U6	50	50	有
U7	60	6	有

业务系统A 数据

合作行

ID 证件号 电话号	X1 资金来源和 经营范围不符 笔数	X2 大额交易 笔数	Y 表现数据
U8	5	10	有
U9	10	2	有
U10	2	30	有
U11	0	10	有
U12	8	7	有

业务系统B 数据

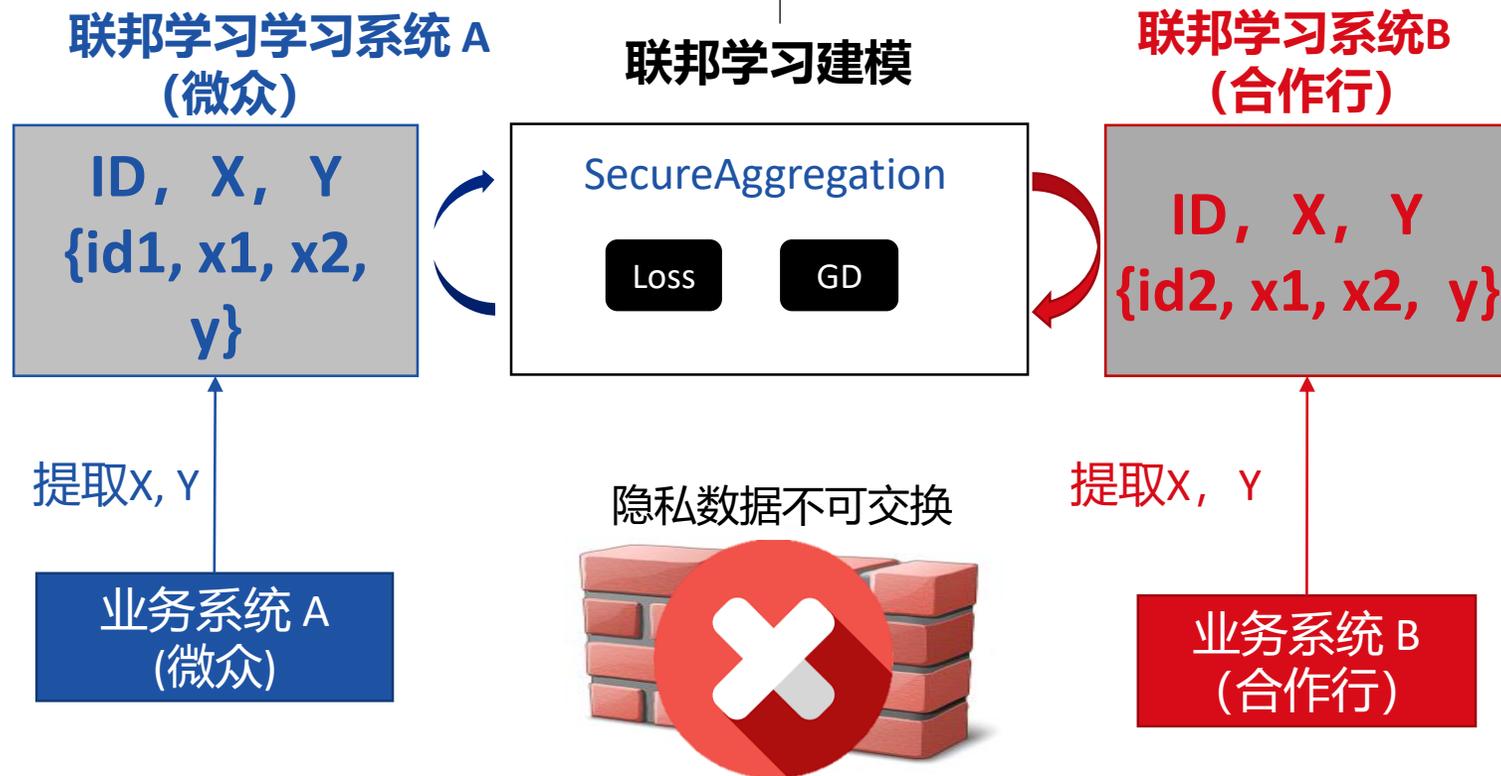
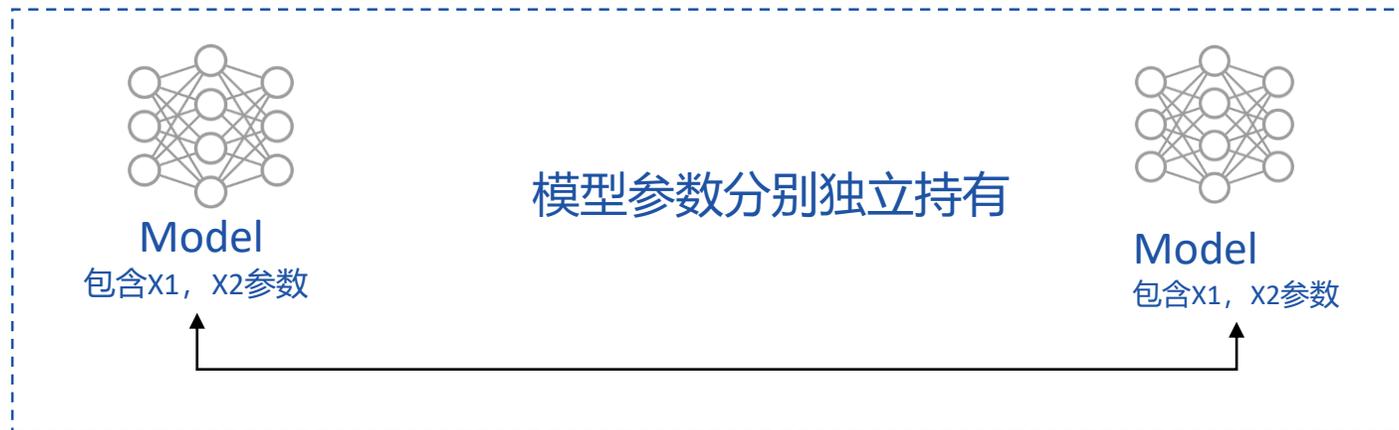
同态加密技术保护隐私

◆ 数据隐私保护:

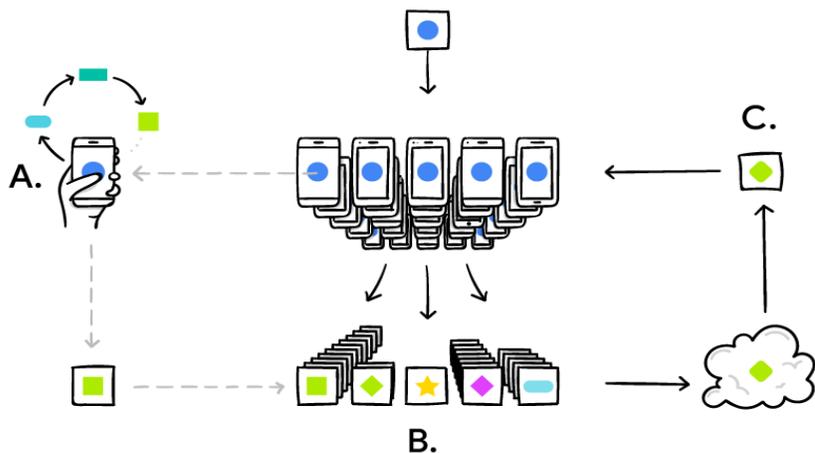
✓ 任何底层X, Y数据不向对方泄露

◆ 结果:

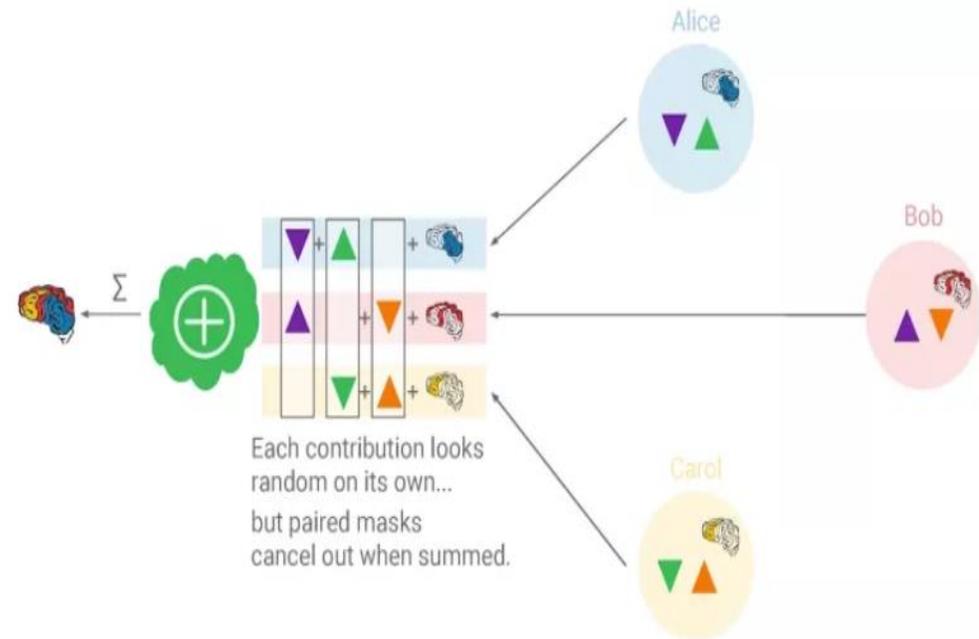
✓ 联合模型比单独建模好



横向联邦核心技术点



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

04

联邦学习开源平台-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

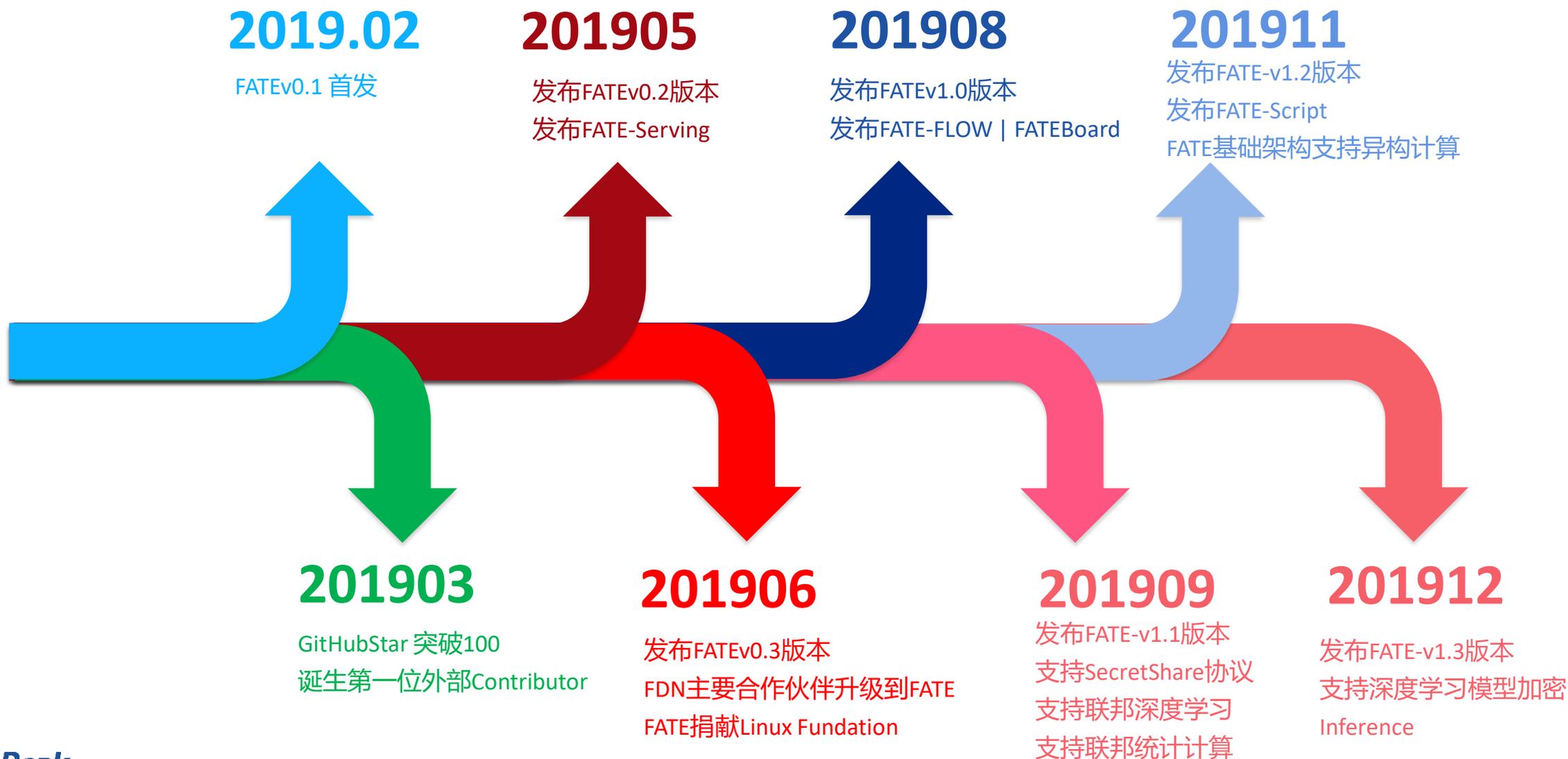
- 支持多种主流算法：为机器学习、深度学习、迁移学习提供高性能联邦学习机制
- 支持多种多方安全计算协议：同态加密、秘密共享、哈希散列等
- 友好的跨域交互信息管理方案，解决了联邦学习信息安全审计难的问题

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



挑战 Challenges in developing a real-world Federated AI

- 一站式建模过程的联邦化
- MPC协议下分布式算法 (on WAN) 易理解和易维护
- 跨站点数据传输安全性和可管理性
- 异构基础架构自适应

技术架构总览

FATE-Flow | FATE-Board

Federated Modeling Visualization

Federated Modeling DAG DSL Parser

Federated Workflow Lifecycle Manager

Federated Task Scheduler

Federated Model Version Manager

FATE-Serving

Processing-app Factory

Online Federated Inference

Online Federated Model
Manager

Dynamic Loaders for model
and processing-app

Platform Suite

Monitor & Alarm

Log Manager

FATE FederatedML: Federated Machine Learning

EggRoll: Distributed Computing & Storage

Federated Network: Cross-Site Networking

Data

Data Access

Data Adapter

HIVE

MySQL

Amazon S3

CSV

Level DB

HBASE

HDFS

.....

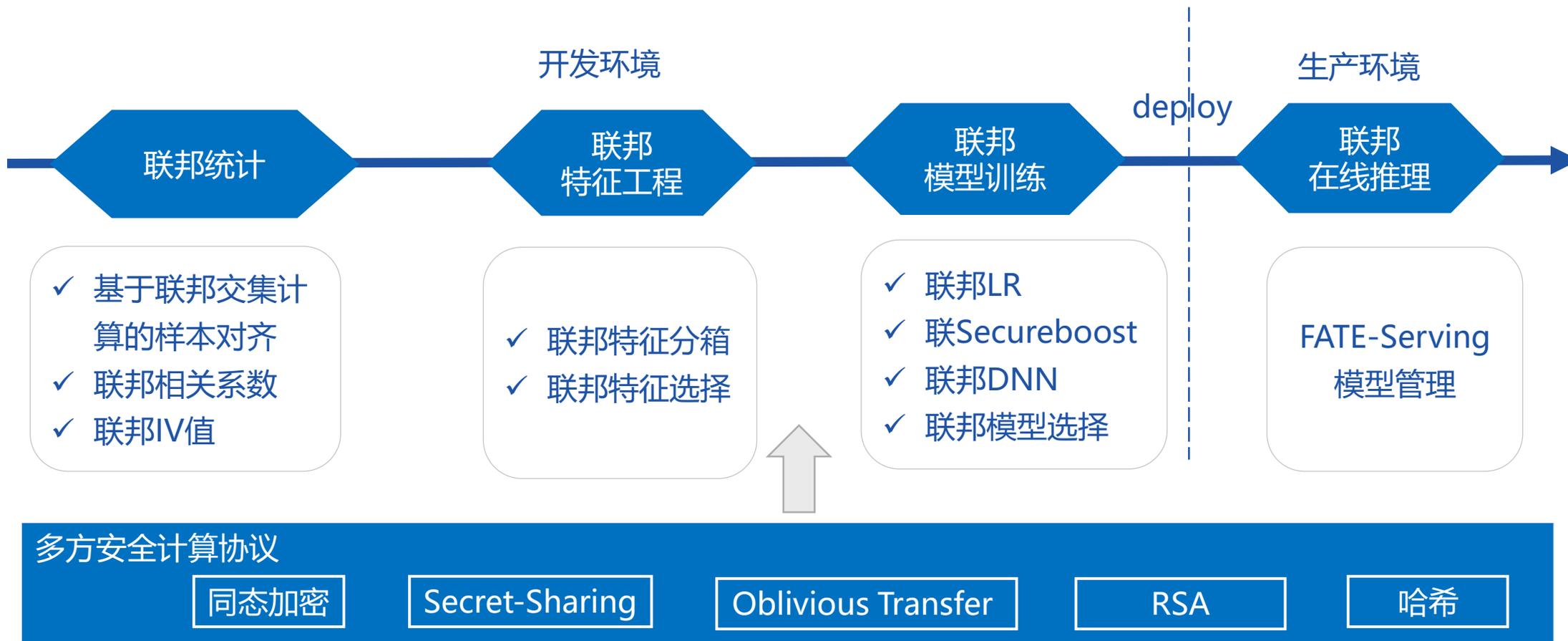
Device

CPU Clusters

GPU Clusters

Andriod / IOS

一站式联合建模Pipeline



核心功能

FATE-Serving



联邦在线模型服务

FATE-Flow | FATE-Board



联邦建模Pipeline和可视化

FATE FederatedML



联邦学习算法各个功能组件

EggRoll



分布式计算和存储抽象

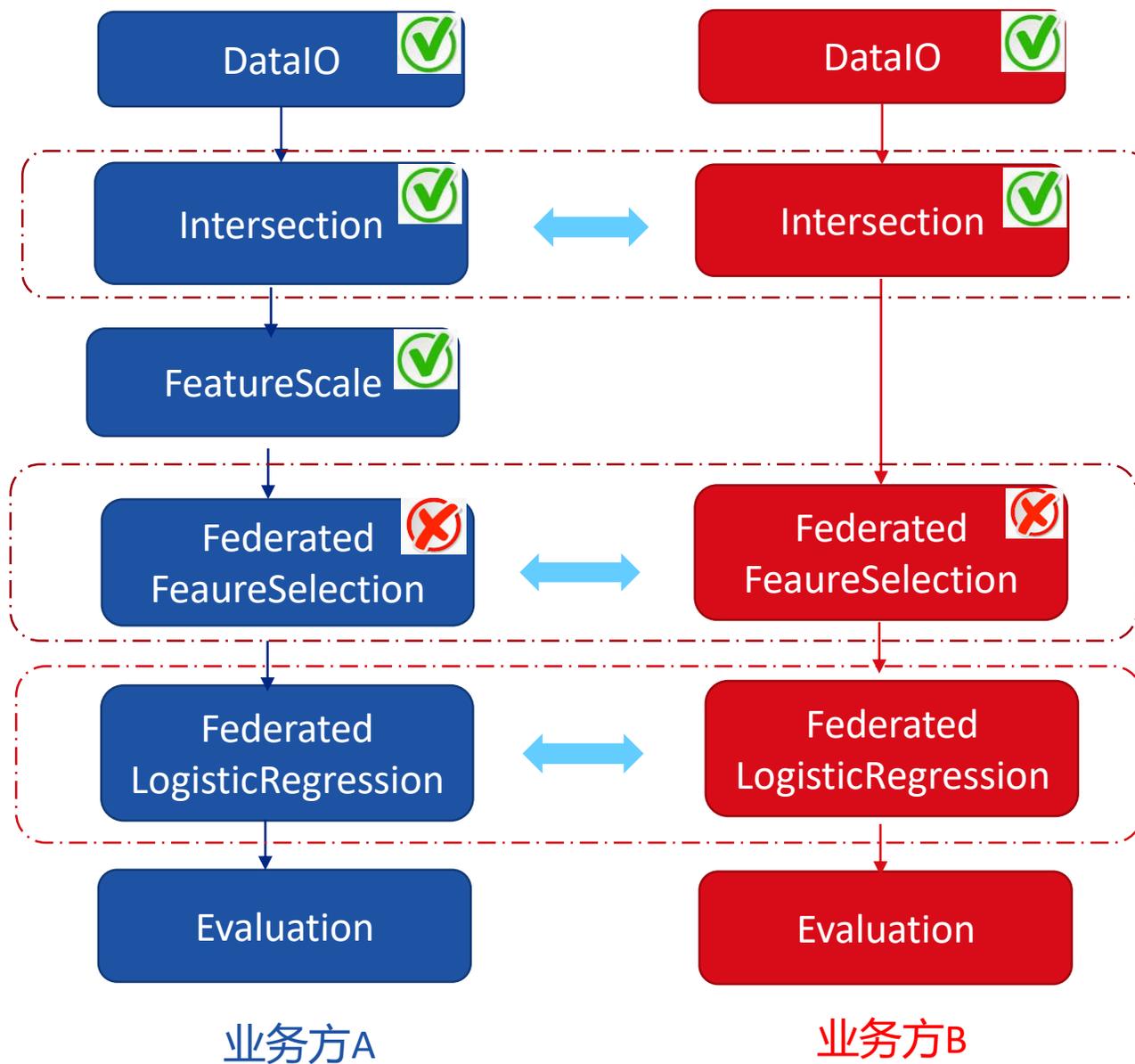
Federated Network



跨站点网络通信抽象

FATE-Flow

- 联邦机制下多方非对称DAG图Paser
- 联邦建模生命周期管理
- 联邦建模实验管理
- 联邦建模模型管理
- 联邦多方任务调度



FATE-Board



FATEBoard Running Jobs

Job: P0001E0002T0001

DATASET INFO.

GUEST	DATASET
10000	lib1.table1 lib1.table2
HOST	DATASET
10000	libh.beacondata1 libh.beacondata2
ARBITER	10000

JOB

elapsed: 00:02:33

62.11 %

kill

[view this job](#)

graph

```

graph TD
    dataio_0 --> intersection_0
    intersection_0 --> federated_sample_0
    federated_sample_0 --> hetero_feature_binning_0
    hetero_feature_binning_0 --> hetero_feature_selection_0
    hetero_feature_selection_0 --> hetero_lr_0
    hetero_lr_0 --> evaluation_0
        
```

LOG

error warning info 14720 debug

```

1 "2019-07-11 17:40:28.318 - job_controller.py[line:290] - INFO: run task 20190711174027288221_1_dataio_0 guest 10000"
2 "2019-07-11 17:40:28.319 - job_controller.py[line:291] - INFO: [DataIOParam]: {input_format: 'dense', delimiter: ',', data_type: 'float64', tag_with_value: False, tag_value_delimiter: '', missing_fill: True, default_value: 0, missing_fill_method: None, missing_input: None, outlier_replace: True, outlier_replace_method: None, outlier_impute: None, outlier_value: 0, 'with_label': True, label_idx: 0, label_type: 'int', output_format: 'dense', initiator: {}, role: 'guest', party_id: 10000}, {guest: [10000], host: [10000], arbiter: [10000]}, config: {data/projects/fate/python/fate_flow/examples/het_hetero_lr_job_conf.json, ds: examples/het_saloon_ds.json, function: 'submitJob', job_parameters: {'model_id': 'arbiter-10000@guest-10000@host-10000@model'}, local: {role: 'guest', party_id: 10000}, CodePath: federatedml/util/data_io.py/DataIO/'module': 'DataIO'}"
3 "2019-07-11 17:40:28.319 - job_controller.py[line:292] - INFO: [data: [args: train_data]]]"
4 "2019-07-11 17:40:28.319 - job_controller.py[line:293] - INFO: [data: [args: [data: -arch.apl.standalone.eggroll_DTable object at 0x7f97a6136080]-]]]"
5 "2019-07-11 17:40:28.319 - model_base.py[line:53] - DEBUG: need_run: True, need_cv: False"
        
```

FATEBoard Running Jobs

Job Overview > Dashboard

JOB SUMMARY dashboard

Job ID	20199717161834519179_2	Role	Guest	Submission Time	2019-04-15 22:40:00
Status	Complete	Party_ID	10000	Start Time	2019-04-15 22:46:00
		Host	5 view	End Time	2019-04-15 22:46:53
		Arbiter	10000	Duration	00:00:53

OUTPUTS FROM JOB

Main Graph
click component to view details

```

graph TD
    dataio_0 --> intersection_0
    intersection_0 --> federated_sample_0
    federated_sample_0 --> hetero_feature_binning_0
    hetero_feature_binning_0 --> hetero_feature_selection_0
    hetero_feature_selection_0 --> hetero_lr_0
    hetero_lr_0 --> evaluation_0
        
```

Parameters(9)

```

compress_thres: 10000
local_only: false
bin_num: 10
method: quantile
head_size: 10000
adjustment_factor: 0.5
display_result: true
error: 0.001
need_run: true
process_method: left
cols: -1
transform_param: {}
transform_type: bin_num
transform_cols: null
        
```

[view this job](#)



FATE FederatedML

Algorithms	Secure Intersection	Secure Federated Feature Engineering	Secure LR	Secure Boost	Secure DNN/CNN	Secure FTL		
ML Operator	Federated Aggregator	Activation	Regulation	Loss	Optimizer	Gradient	Hessian	
Numeric Operator	Add	Sub	MUL	DIV	Comparison	AND	OR	Scalar Product
MPC Protocol	Homomorphic Encryption	Secret-Sharing	Oblivious Transfer	Garbled Circuit	RSA			
Eggroll & Federation API	Map	MapPartitions	MapValues	Reduce	Join	Remote	Get	

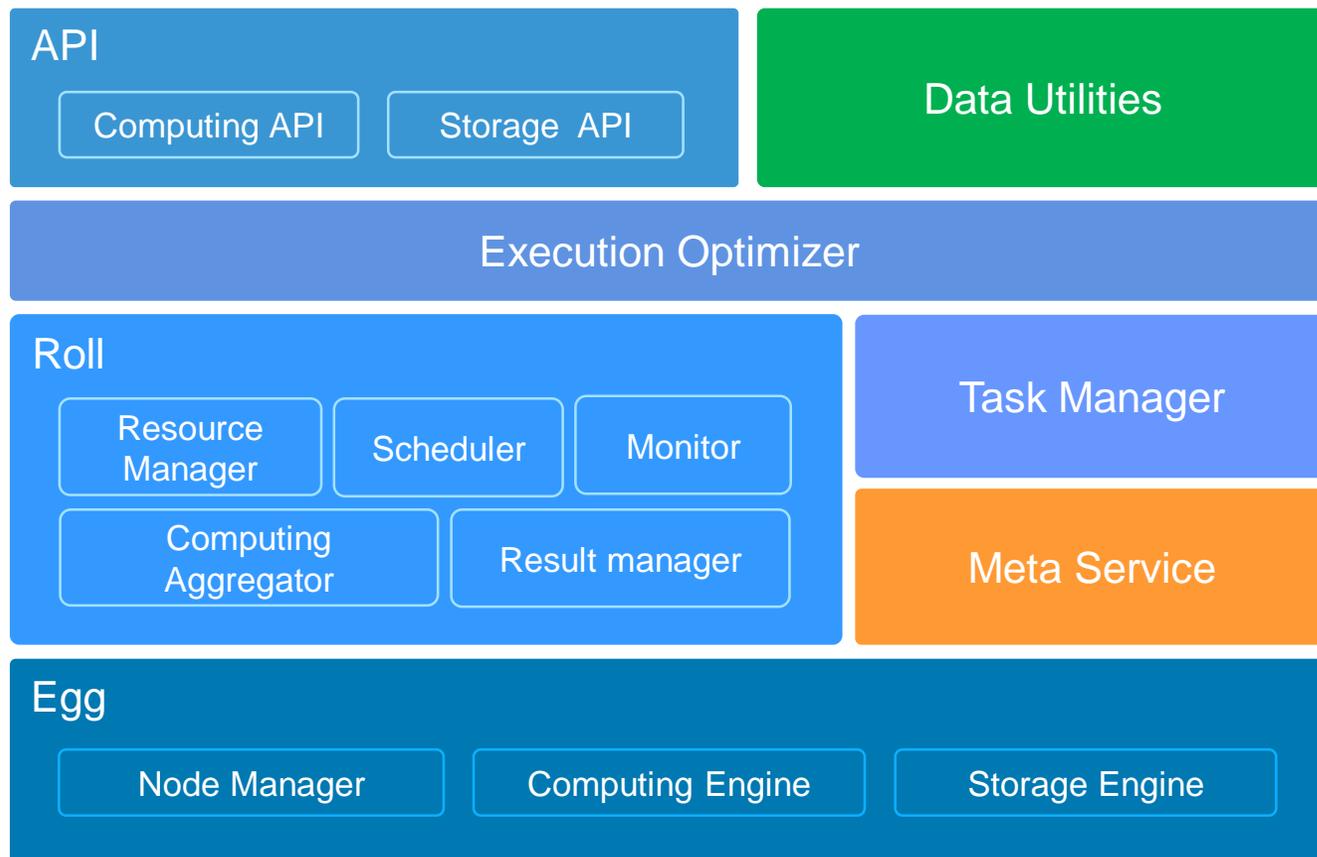
EggRoll

- 编程框架

- EggRoll API: 面向算法开发者, 通过API实现分布式计算.

- 计算/存储架构

- 联邦学习一方分布式计算和存储
- 模块
 - Meta-Service: 元信息管理
 - Roll: 数据 / 计算 调度 (to eggs), 聚合操作等
 - Egg: 计算、存储引擎



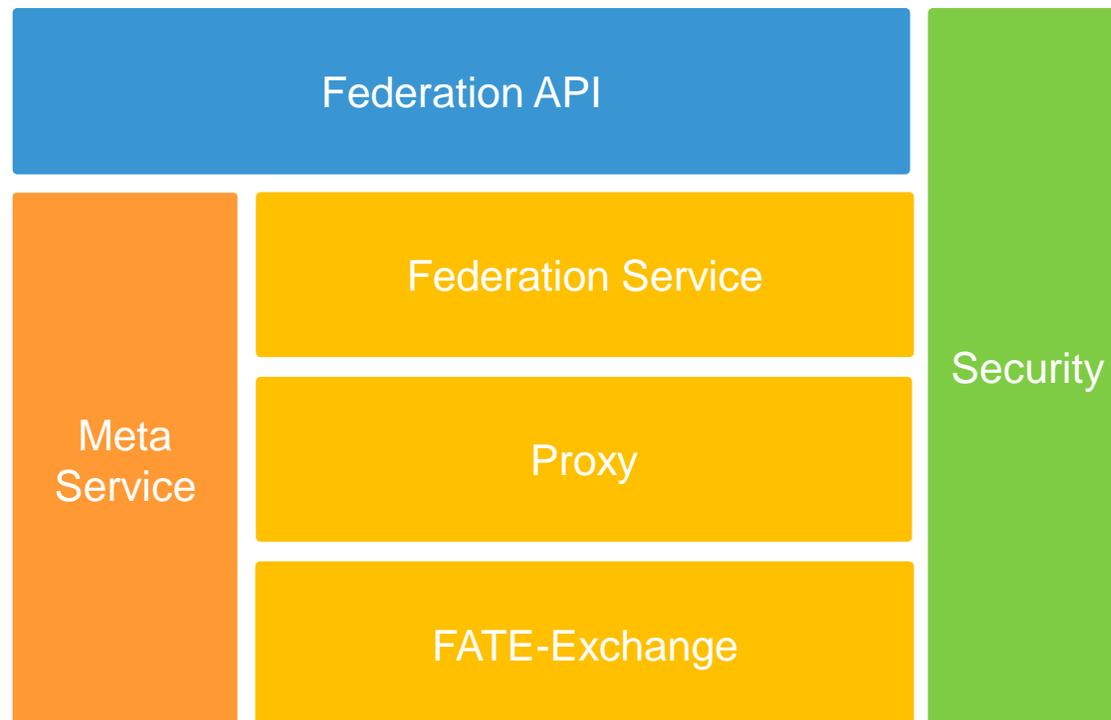
Federated Network

- **编程框架**

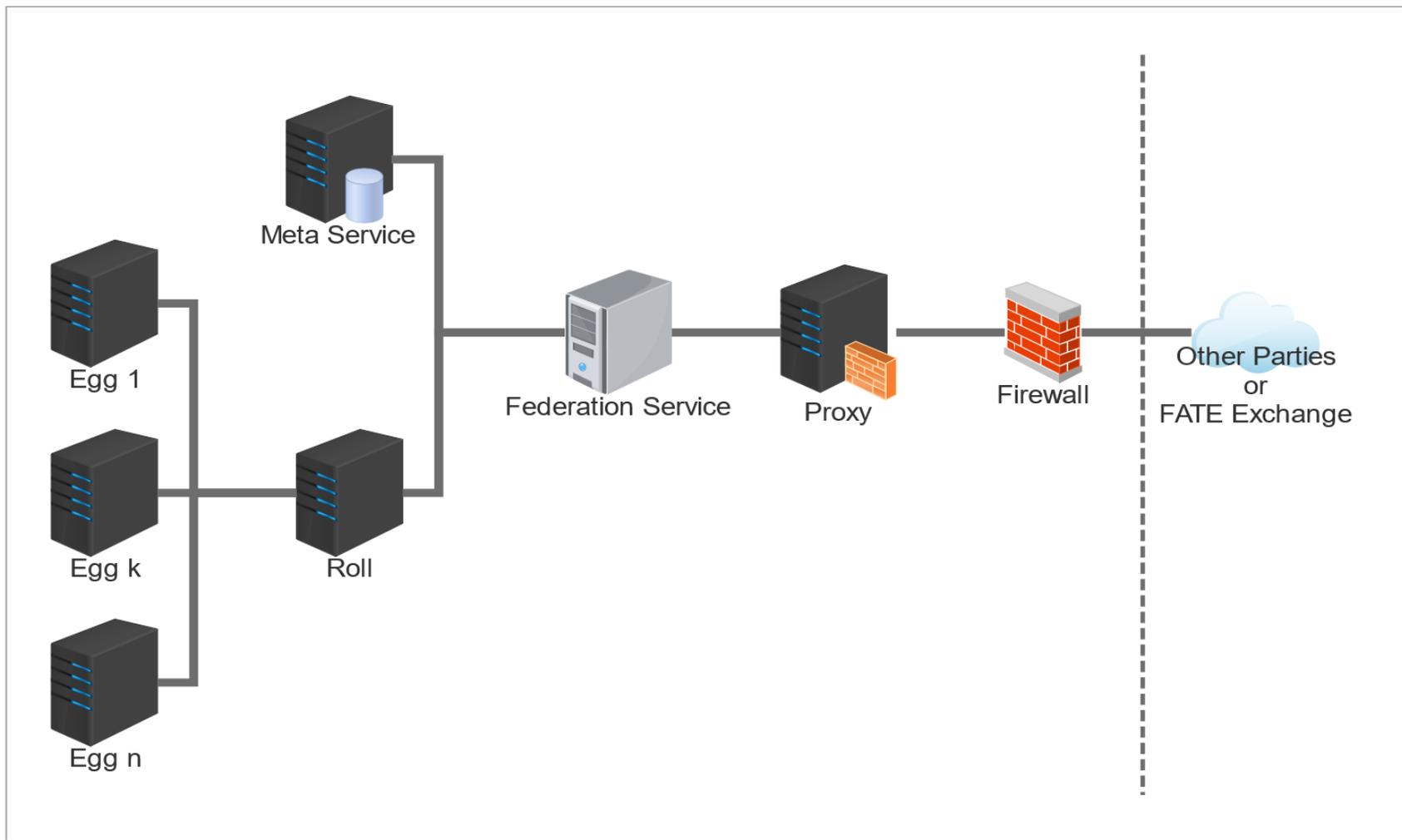
- Federation API: 面向算法开发者, 通过API实现跨站点通信和交互.

- **通信架构**

- 联邦学习多个参与方跨站点通信
- 模块
 - Meta-Service: 元信息管理
 - Proxy: 应用层联邦学习路由
 - Federation: Global Object (i.e. data to be 'federated' among parties) 抽象和实现
 - FATE-Exchange

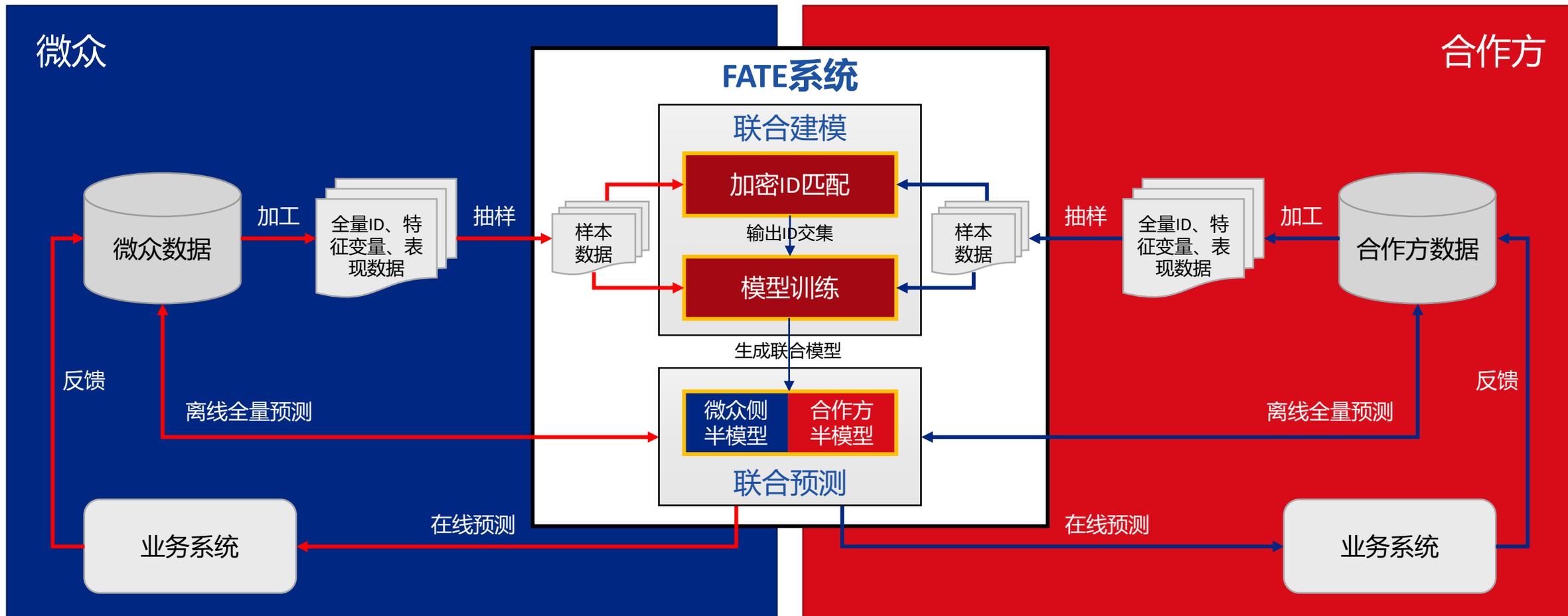


一方部署网络拓扑-示例



基于FATE的联合建模

联合建模、预测示意图 —— 安全合规的数据合作过程



开发流程

1

选择一个机器学习算法,
设计多方安全计算协议



2

定义多方交互的数据变
量



3

构建算法执行工作流



4

基于EggRoll &
Federation Api 实现算
法工作中各个功能
组件



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

关注FATE



FATE GitHub主页



FATE 微信小助手

Join FATE, Let's Federated Everything!

官网: <https://www.fedai.org/>

邮箱: contact@fedai.org