海星(HiStar)——联邦学习中间件

https://git.openi.org.cn/PCL Federated.Learning.Middleware/HiStar

目录：

- 背景介绍
- HiStar设计理念
- HiStar开源现状
- 生态和未来发展规划

目录：

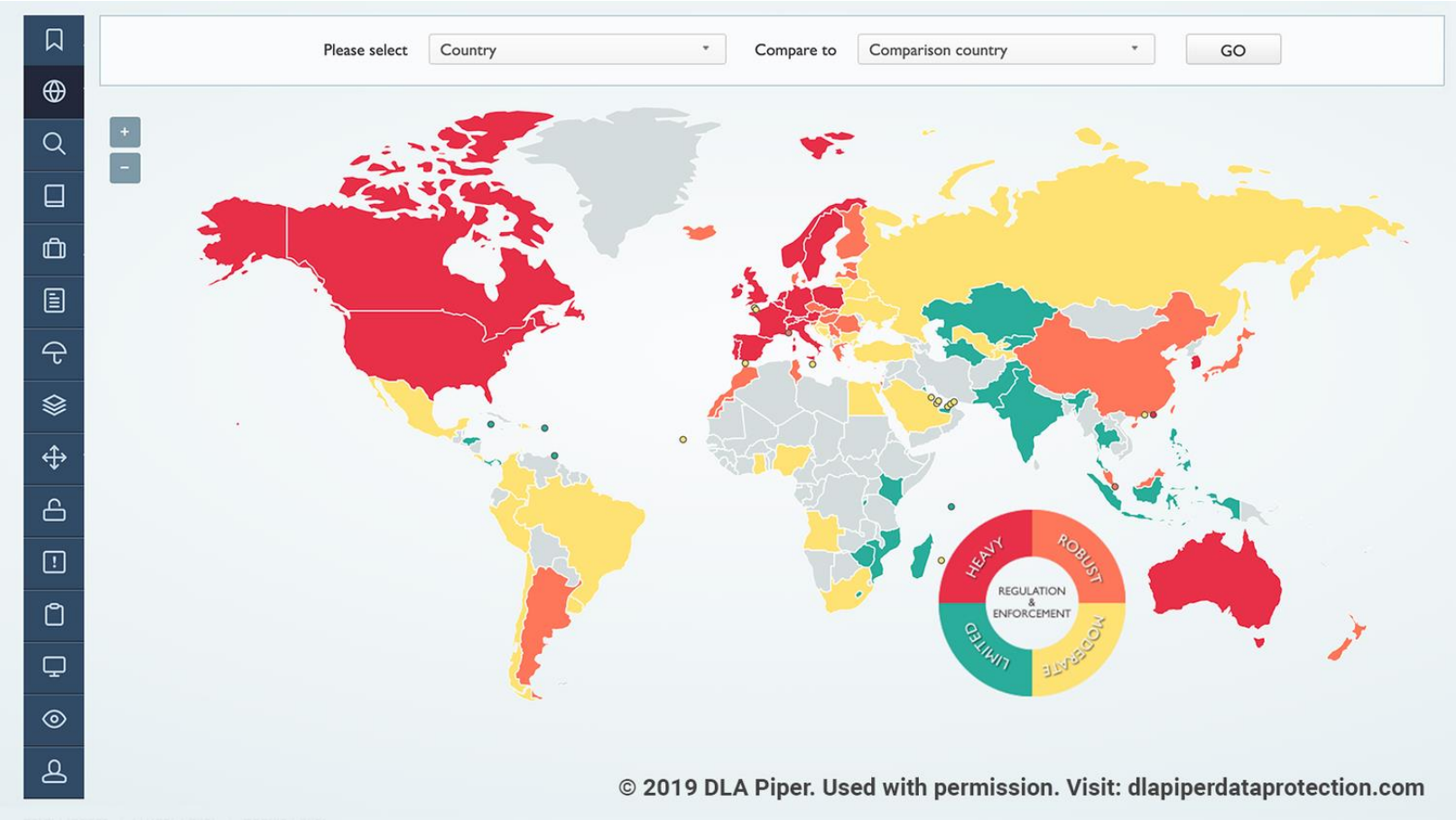
- 背景介绍
- HiStar设计理念
- HiStar开源现状
- 生态和未来发展规划

隐私保护机器学习兴起的原因

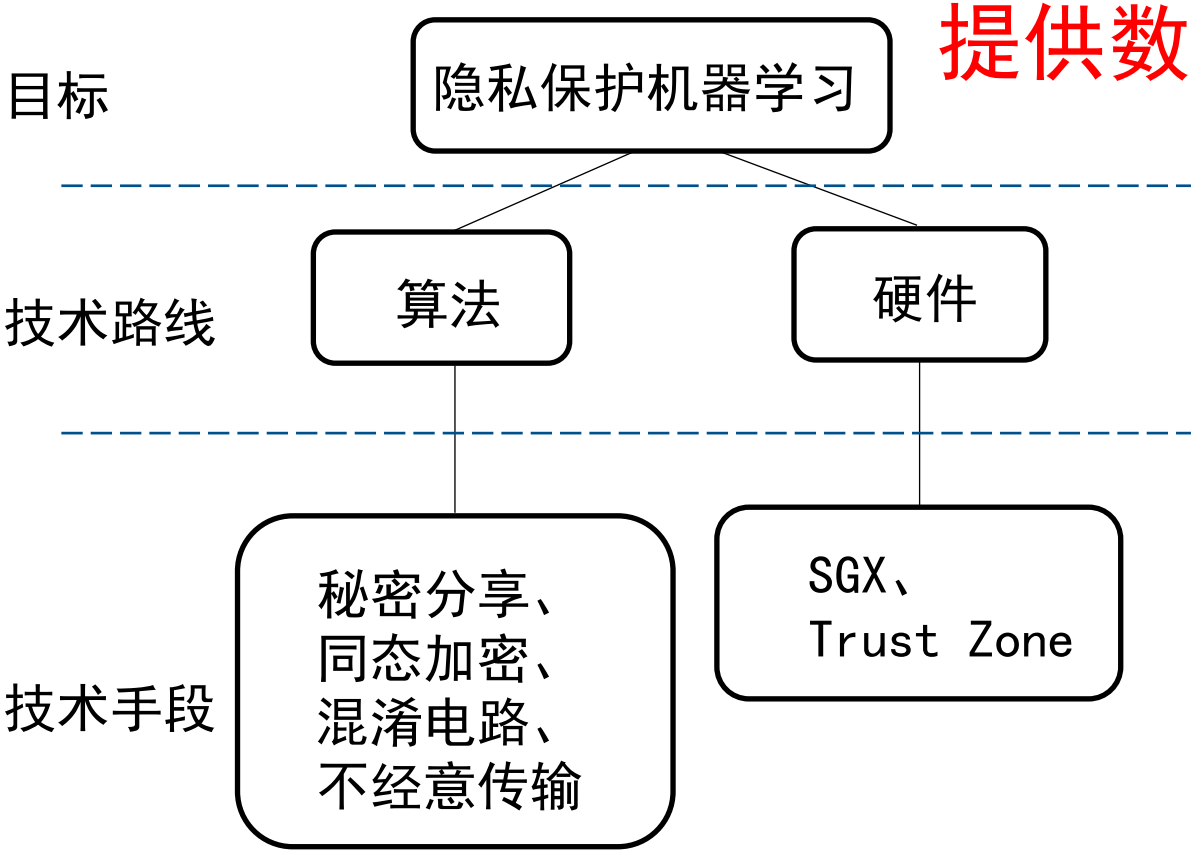
数据安全、个人隐私保护等问题受到社会广泛关注，相关法律法规相继出台或公布：

- 欧盟：《一般数据保护条例》；
- 中国：《民法典》、《出口管制法》、《数据安全法》、《个人信息保护法（草案）》 等

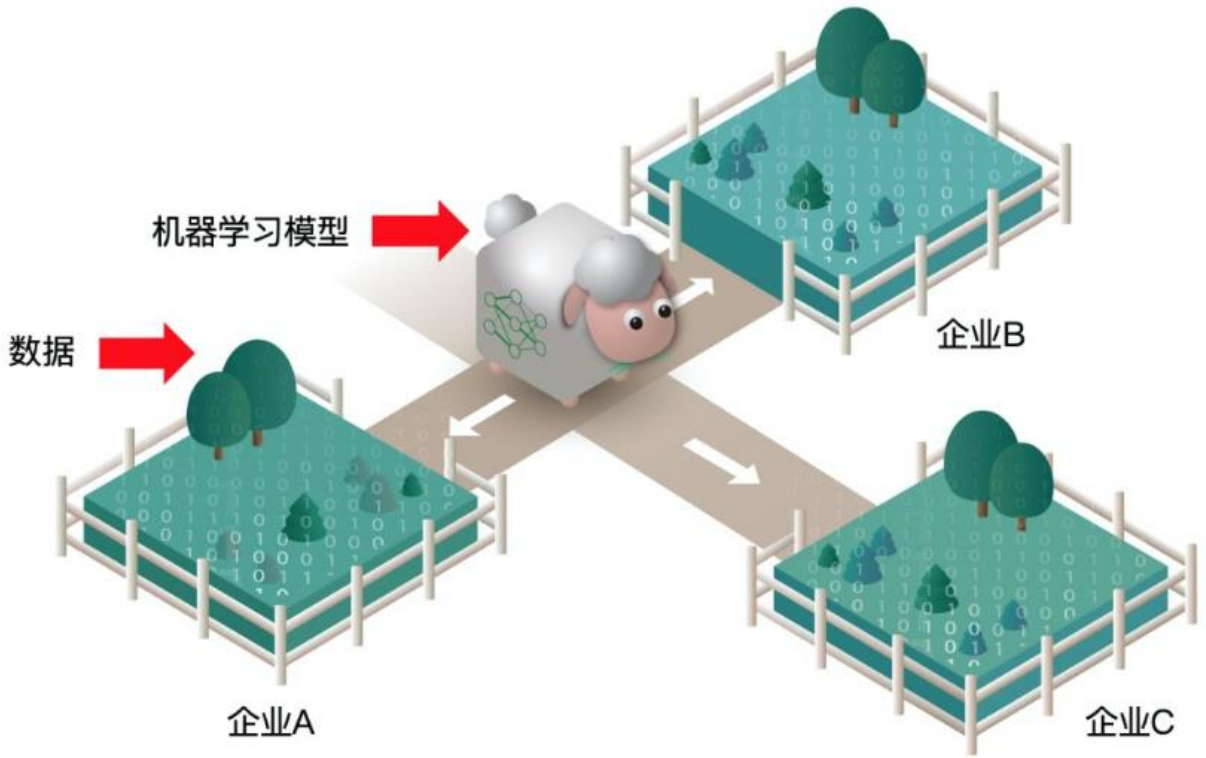
如何在不侵犯隐私数据的情况下，实现数据价值的共享是机器学习领域亟需解决的问题！



隐私保护机器学习现有技术



提供数据使用权，保留数据所有权。



数据可用而不可见，充分释放数据价值！

联邦学习场景分类

联邦机器学习是一个机器学习框架，能有效帮助多个机构在满足用户隐私保护、数据安全和政府法规的要求下，进行数据使用和机器学习建模。

目录：

- 背景介绍
- HiStar设计理念
- HiStar开源现状
- 生态和未来发展规划

HiStar设计理念

- **非侵入式：** 深度学习算法的开发人员在使用HiStar将深度学习算法转换为具有隐私保护的联邦深度学习算法时无需侵入原有算法，如将深度学习中的非线性运算替换为多项式进行计算。

- **多场景适用：** 使用HiStar可以针对多种不同的机器学习使用场景进行隐私保护方案的定制，既能满足数据横向切分/纵向切分的训练与推理，也可满足集中式/分布式模型的训练与推理。

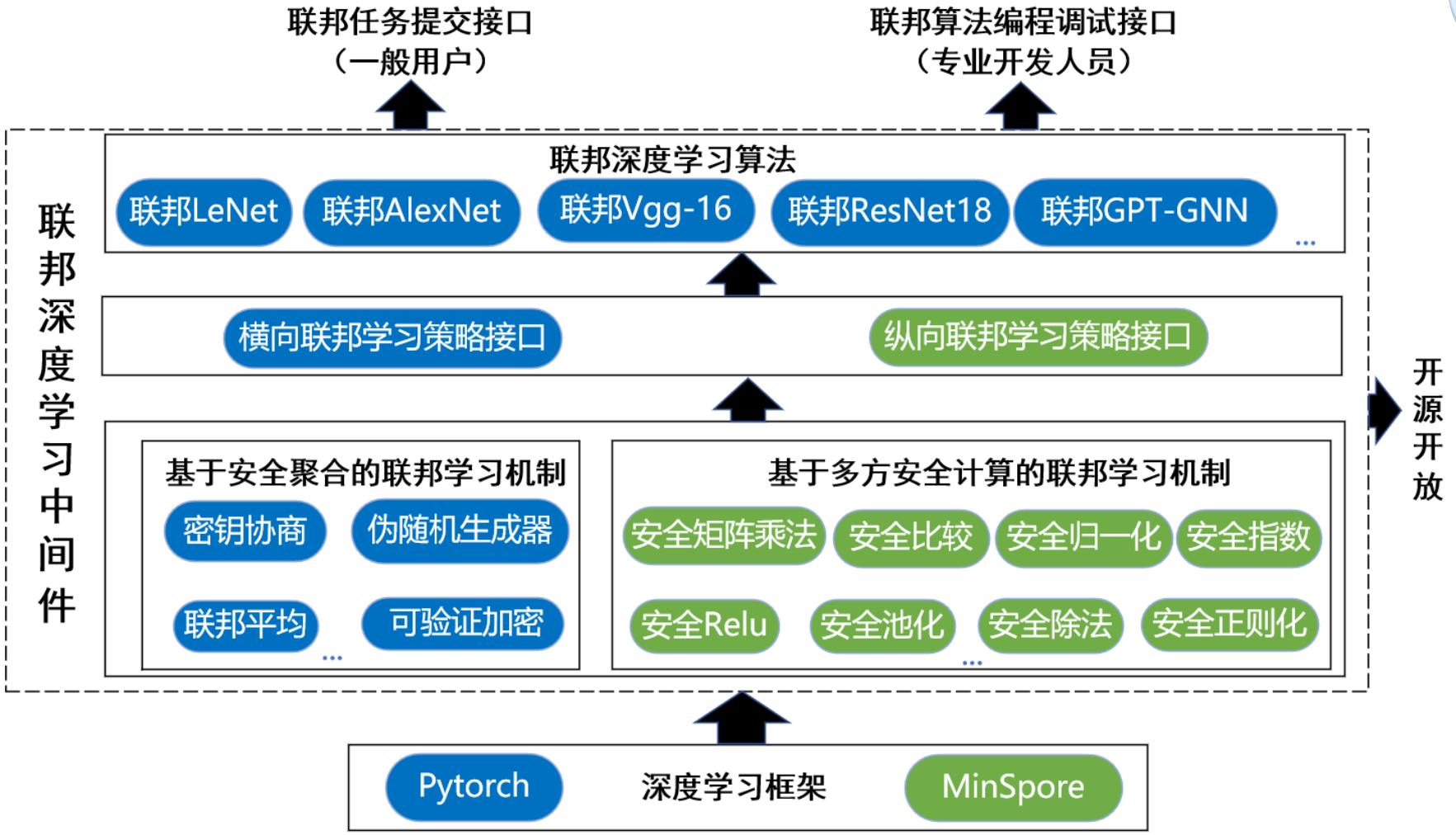
- **适用于端-边-云应用架构：** HiStar将兼容以“云”为主导进行大规模神经网络预训练；以“云”及“边”为参与方进行联邦微调；通过“端边云”协同进行联邦推理的应用架构。

- **性能-安全综合调控：** HiStar将融合多种隐私保护策略，实现多种隐私保护手段，并对性能及安全综合考量与评测，可基于不同业务场景灵活选择合适的安全技术。

HiStar整体架构

已具备功能：

待开发功能：

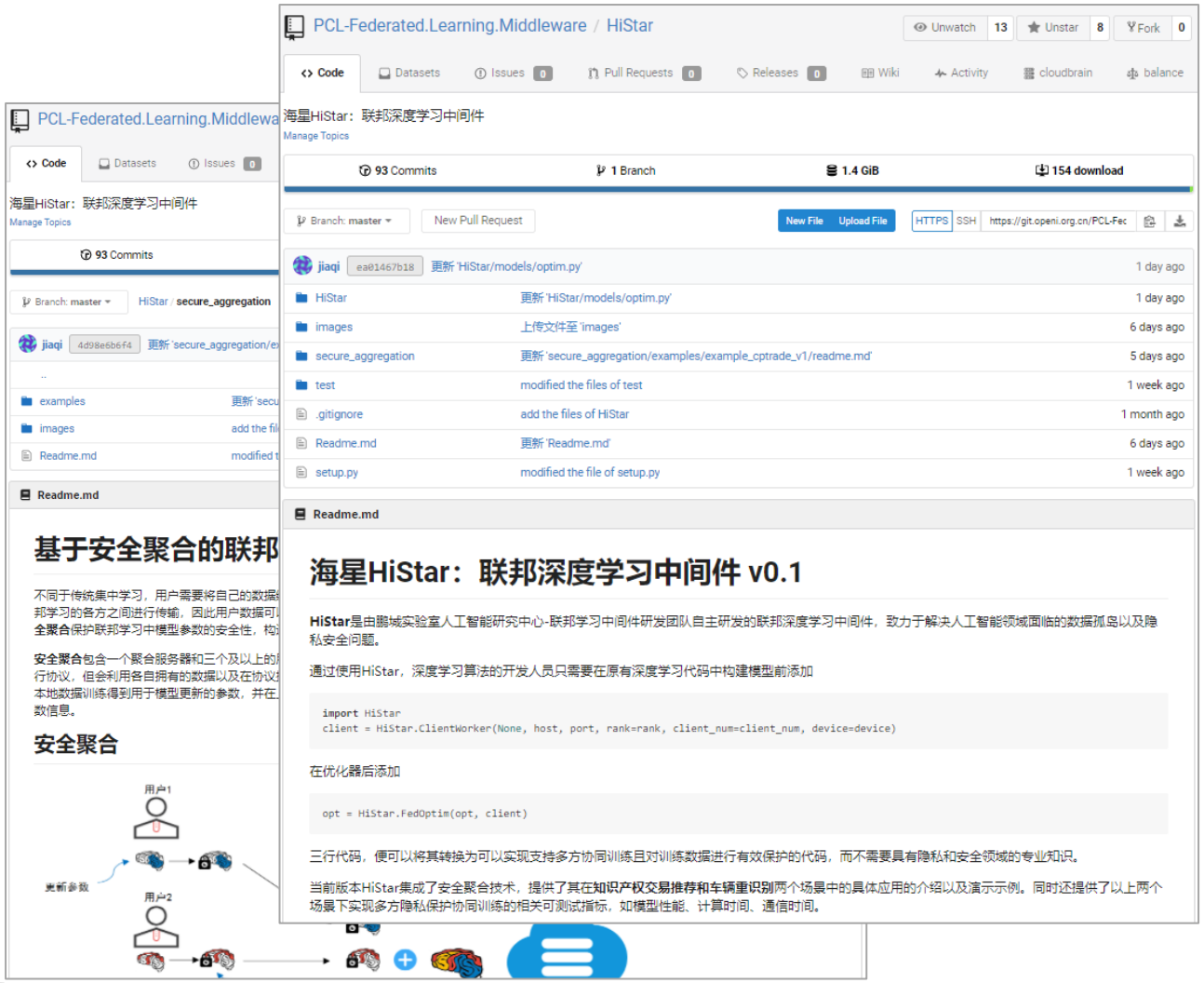


目录：

- 背景介绍
- HiStar设计理念
- HiStar开源现状
- 生态和未来发展规划

HiStar——Openl开源

Openl社区开源网页：

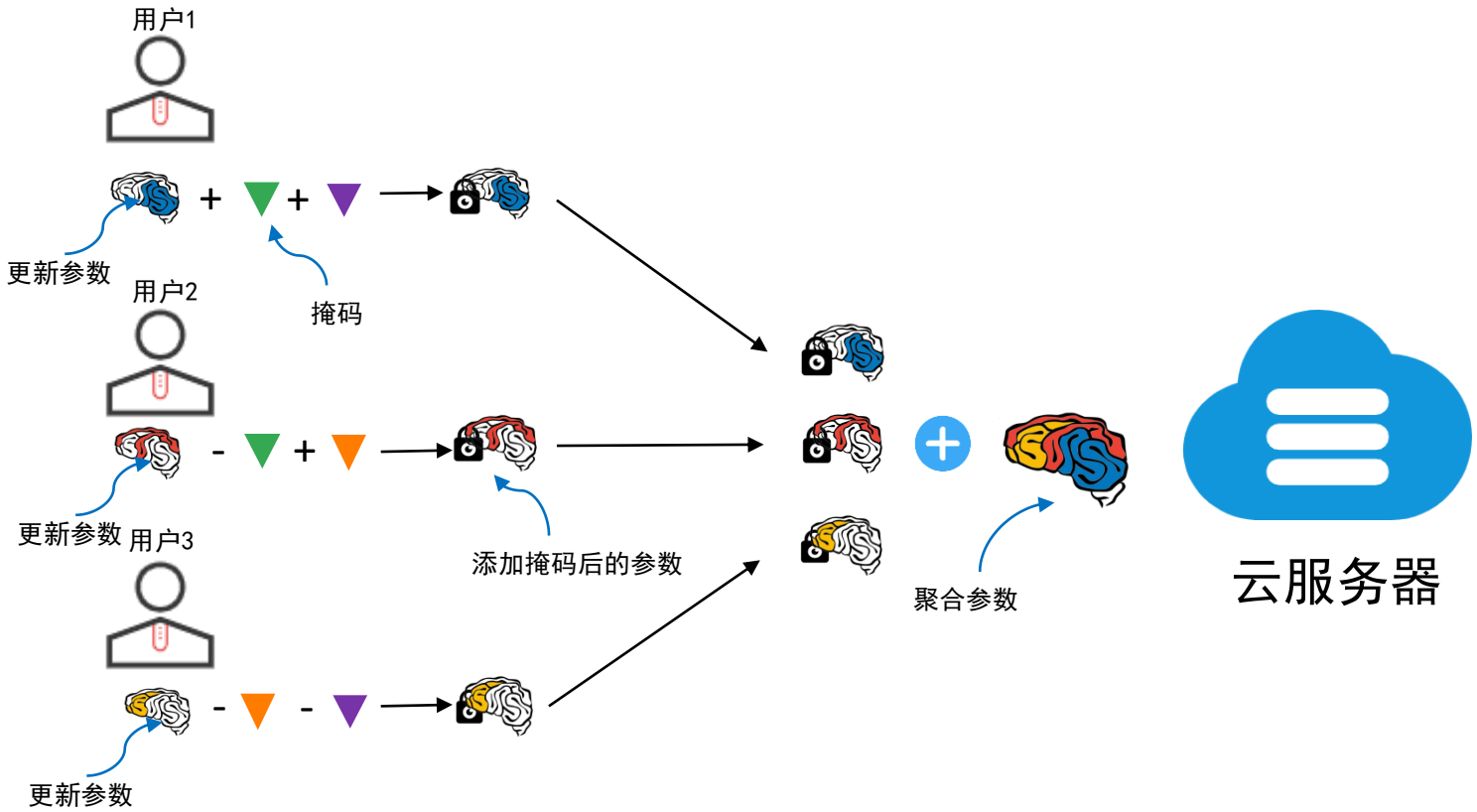


微信交流群：

HiStar联邦学习中间件交流群

该二维码7天内(7月2日前)有效，重新进入将更新

基于安全聚合的联邦学习机制



基于安全聚合的联邦学习机制设计流程

1. Server_initialization()
   功能：服务器与用户建立网络通信，建立用户的编号列表，加入安全聚合协议

2. Client_initialization()
   功能：用户与服务器建立网络通信，加入安全聚合协议

3. Client_key_generation()
   功能：通过聚合协议的（G,g,q,H）生成DH密钥对(secret_key, public_key)

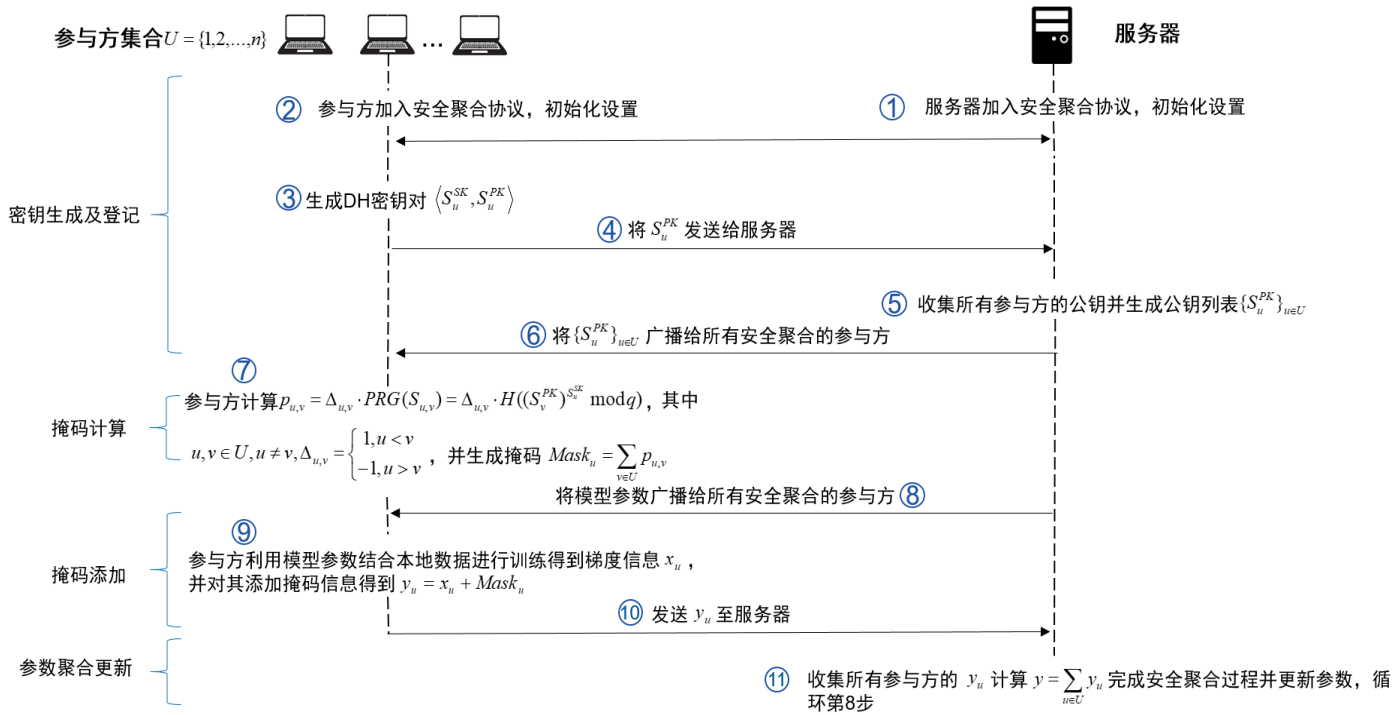
4. Client_send(server, public_key)
   功能：发送公钥

5. Server_gather(clients, public_keys)
   功能：收集用户公钥并创建公钥列表

6. Server_broadcast(clients, public_keys)
   功能：将公钥列表广播给用户

7. Client_mask_generation(secret_key, public_keys)
   功能：通过聚合协议的PRG以及自己私钥和公钥列表产生掩码mask

8. Server_broadcast(clients, model_paras)
   功能：广播模型参数给用户



9. Client_epoch(data, model_paras，mask)
   功能：用户用自己数据进行模型训练更新参数，并加入掩码得到掩码后的参数y_mask

10. Client_send(server, y_mask)
   功能： 用户发送掩码后的参数给服务器

11. Server_gather(clients, y_masks)
   功能： 服务器收集用户掩码参数y_masks

HiStar运行

深度学习算法的开发人员在使用HiStar将现有的深度学习算法转换为具有隐私保护的联邦学习算法发只需要在原有深度学习算法代码中模型构建部分之前添加

```
import HiStar
client = HiStar.ClientWorker(None, host, port, rank=rank, client_num=client_num, device=device)
```

在优化器代码之后添加

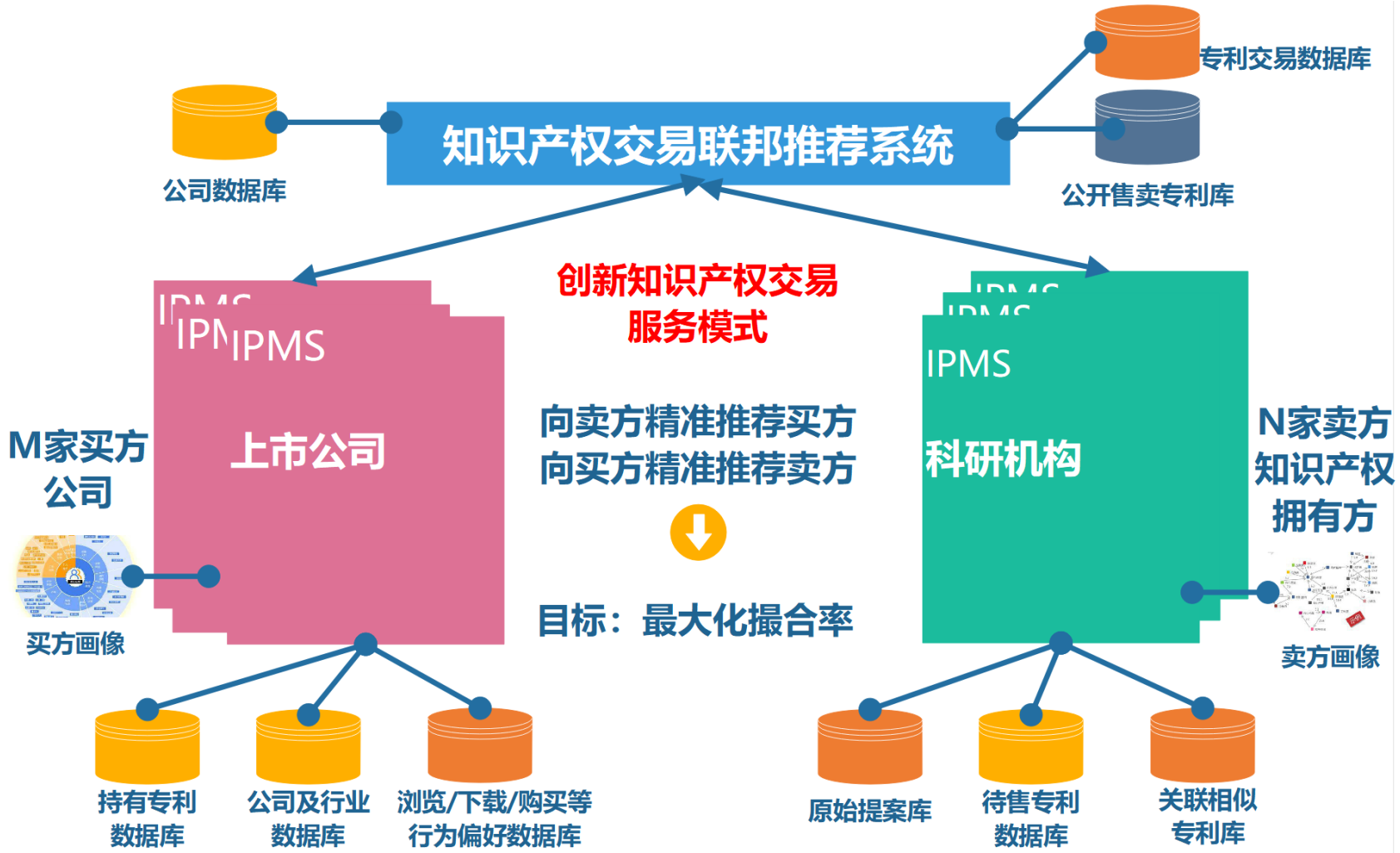
```
opt = HiStar.FedOptim(opt, client)
```

三行代码便可以实现，而不需要具有隐私和安全领域的专业知识。

```
352     optimizer = torch.optim.AdamW(model.parameters(), lr = args.lr)
353     scheduler = torch.optim.lr_scheduler.CosineAnnealingLR(optimizer, 500, eta_min=1e-6)
354
355     writer = SummaryWriter(os.path.join(args.log_dir, args.task_name))
356
357
358     epoch_batch_num = args.n_repeat
```

```
354+    client = HiStar.ClientWorker(None, '0.0.0.0', 9088, RANK, 7, device, verbose=False)
355+
356     optimizer = torch.optim.AdamW(model.parameters(), lr = args.lr)
357     scheduler = torch.optim.lr_scheduler.CosineAnnealingLR(optimizer, 500, eta_min=1e-6)
358+    optimizer = HiStar.FedOptim(optimizer, client)
359
360     writer = SummaryWriter(os.path.join(args.log_dir, args.task_name))
361
362
363     epoch_batch_num = args.n_repeat
```

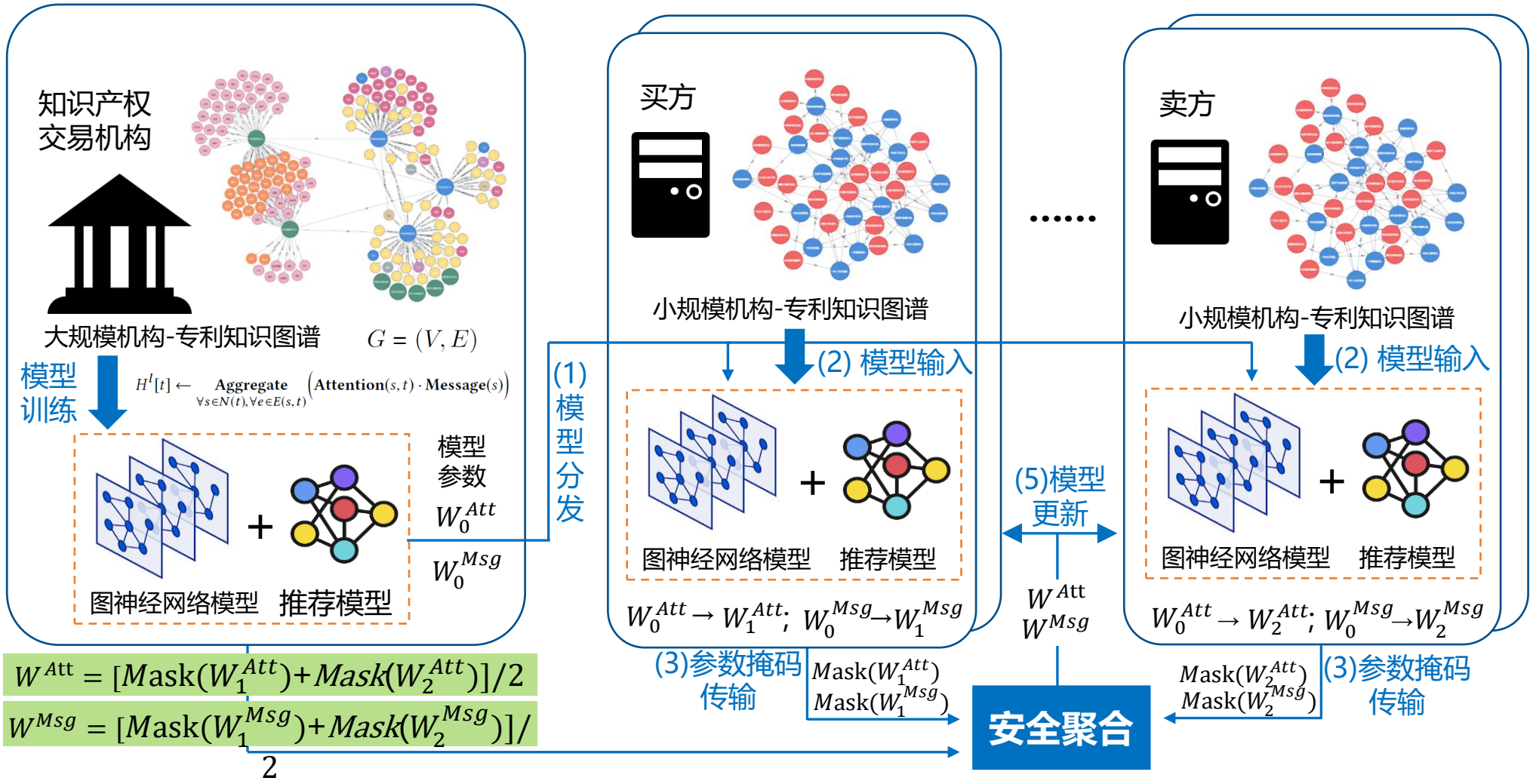
在知识产权交易场景中的应用



训练阶段

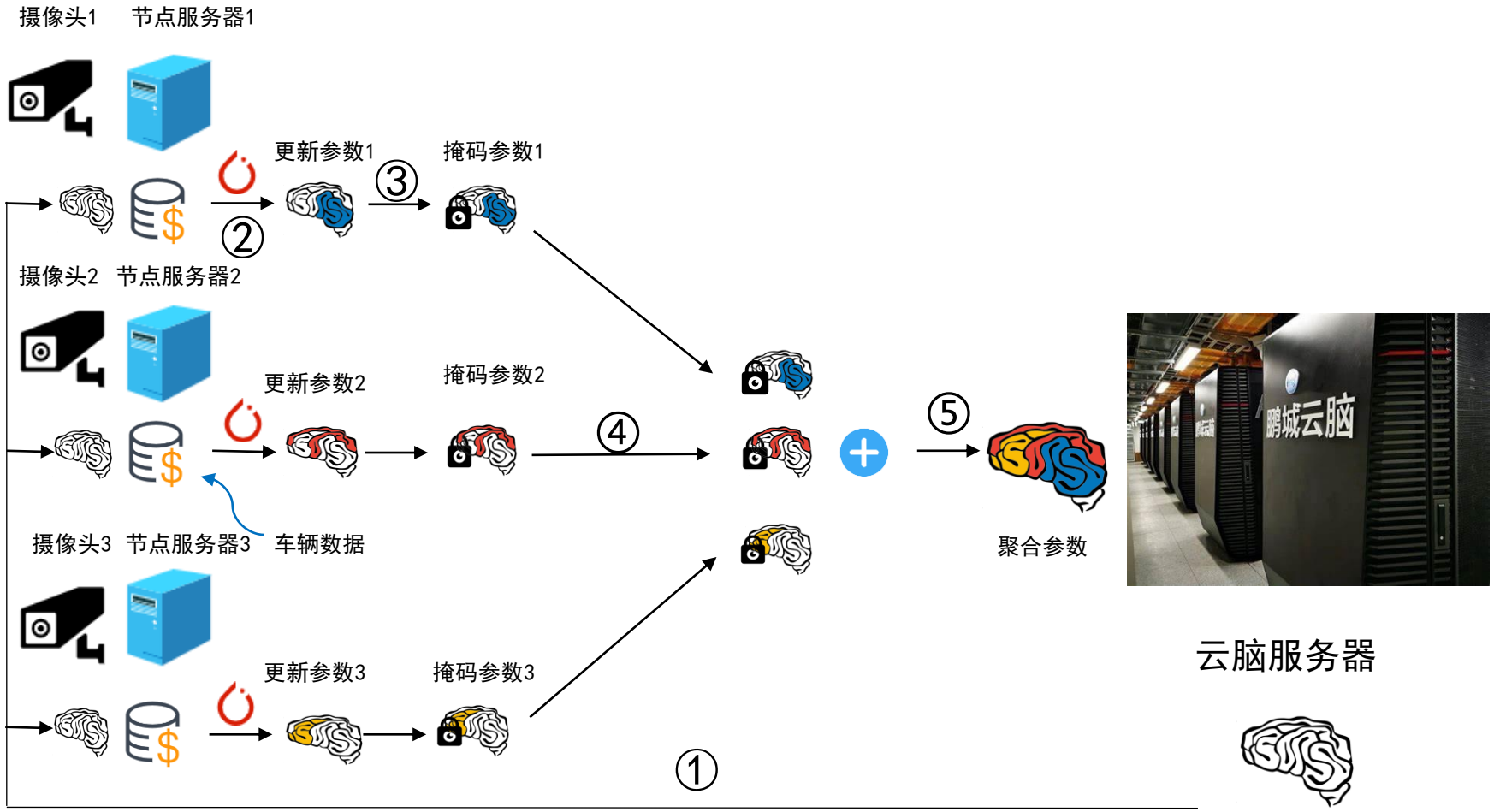
① 预训练阶段

② 微调阶段



(4) 协作聚合：微调模型在知识产权交易机构完成模型平均并分发给买卖双方

在车辆重识别（ReID）场景中的应用



云脑服务器

预训练模型

目录：

- 背景介绍
- HiStar设计理念
- HiStar开源现状
- 生态和未来发展规划

生态和未来发展规划

• 功能完善

- 完善基于安全聚合的联邦学习机制，增加训练过程中用户掉线的场景；
- 增加基于安全多方计算的联邦学习机制。

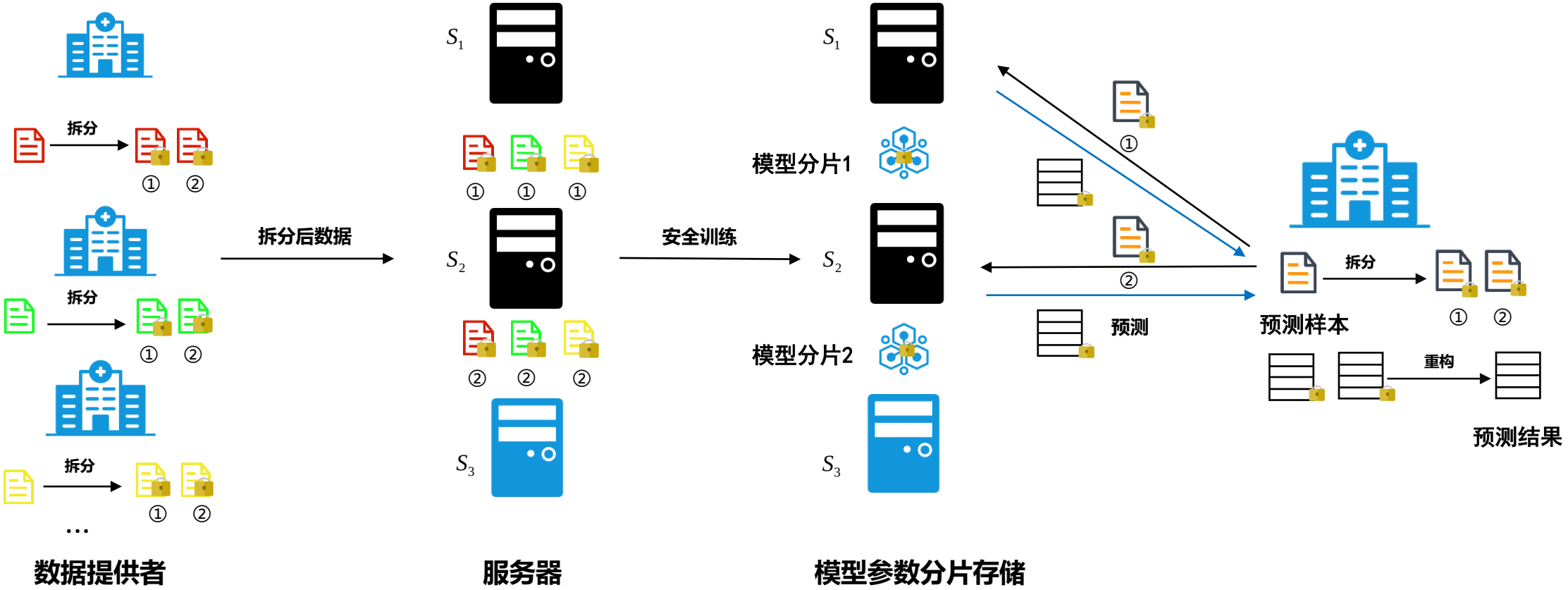
• 场景拓展

- 增加HiStar在智慧医疗、机器翻译等场景中的应用示例；
- 增加在纵向联邦的相关场景的应用示例。

• 对外合作

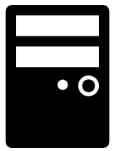
- 与业内知名厂商，开放社区合作共同完成HiStar的开发工作；
- 以HiStar作为解决方案提供给业内厂商，在实际场景进行落地应用。

基于多方安全计算的联邦学习机制

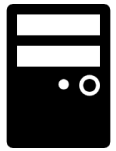


安全加法运算

服务器 S_1



服务器 S_2



服务器 S_3

输入：

$[x]_1^L, [y]_1^L$

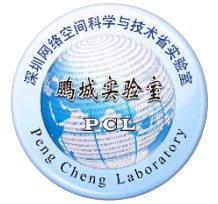
$[x]_2^L, [y]_2^L$

输出：

$[z]_1^L = [x+y]_1^L = [x]_1^L + [y]_1^L$

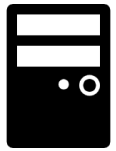
$[z]_2^L = [x+y]_2^L = [x]_2^L + [y]_2^L$

安全乘法运算



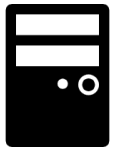
输入:

服务器 S_1



$[x]_1^L, [y]_1^L$

服务器 S_2



$[x]_2^L, [y]_2^L$

服务器 S_3

三元组： $a, b \in_R Z_L, c = ab$

$([a]_1^L, [b]_1^L, [c]_1^L)$

$([a]_2^L, [b]_2^L, [c]_2^L)$

$([a]_1^L, [a]_2^L) \leftarrow a, ([b]_1^L, [b]_2^L) \leftarrow b, ([c]_1^L, [c]_2^L) \leftarrow c$

$[e]_1^L = [x]_1^L - [a]_1^L, [f]_1^L = [y]_1^L - [b]_1^L$

$[e]_2^L = [x]_2^L - [a]_2^L, [f]_2^L = [y]_2^L - [b]_2^L$

$e = [e]_1^L + [e]_2^L = x - a,\ f = [f]_1^L + [f]_2^L = y - b$

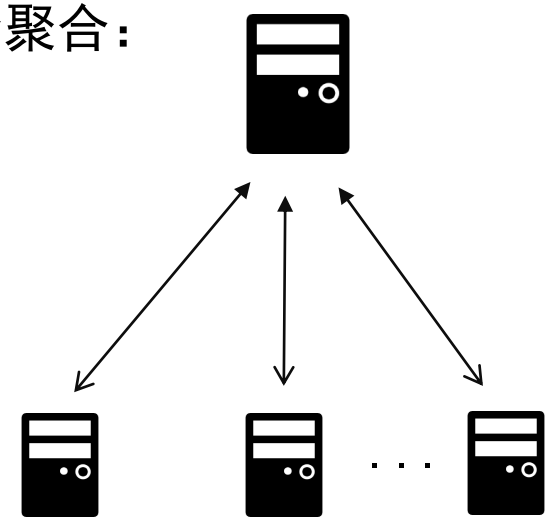
输出:

$[z]_1^L = [xy]_1^L = [x]_1^L f + [y]_1^L e + [c]_1^L$

$[z]_2^L = [xy]_2^L = [x]_2^L f + [y]_2^L e + [c]_2^L - ef$

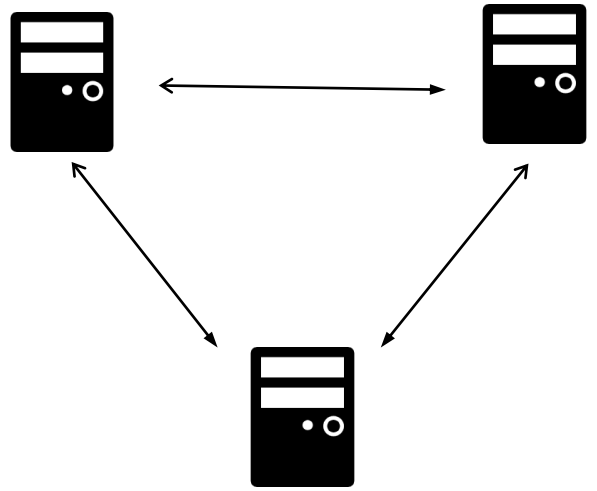
安全聚合机制 VS 多方安全计算机制

安全聚合：



特点：本地明文训练，传输加密梯度

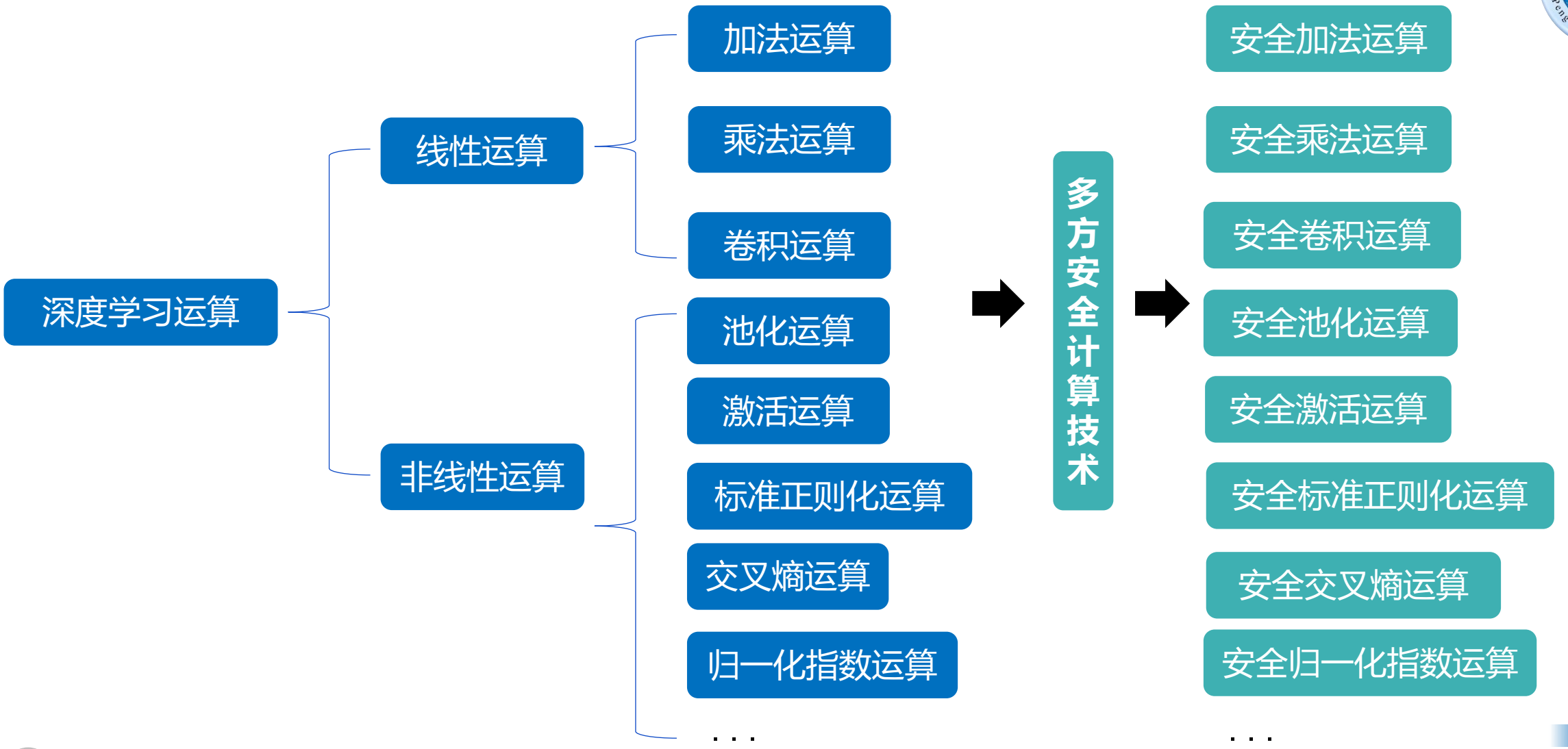
多方安全计算：



特点：多台服务器在私密数据上进行安全训练

联邦机制	支持联邦场景	是否支持模型隐私保护	是否有中心服务器
安全聚合	横向	否	是
多方安全计算	横向/纵向	是	否

基于多方安全计算机制的设计原理



感谢聆听

Q & A