隐私计算
核心技术

目录

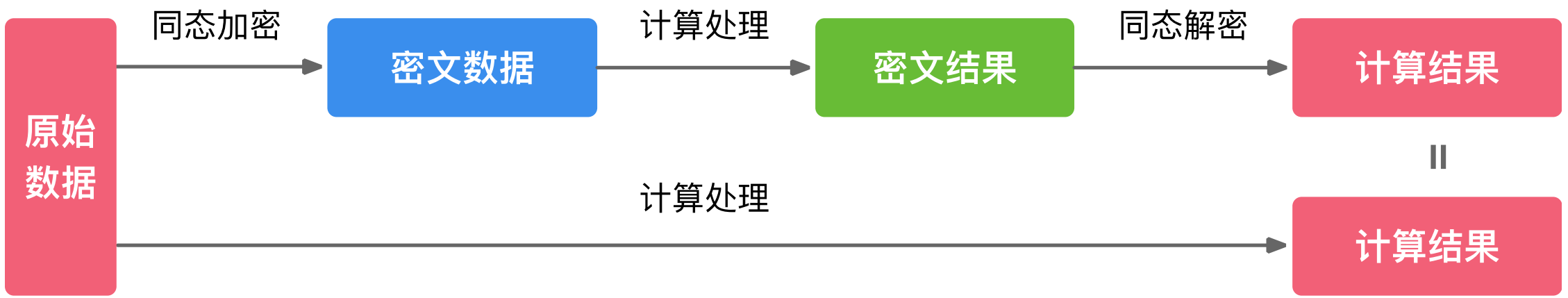
细节1：同态加密

同态加密

- 1978 年，Ron Rivest、 Leonard Adleman 和 Michael L. Dertouzos 提出同态加密问题，并在同年提出满足乘法同态的 RSA 算法。
- 2009 年，Gentry 提出了首个实用的全同态加密算法,标志着全同态计算时代的开始。
- 2017 年，国际同态加密标准委员会成立,标志着同态加密在全球进入高速发展阶段。

同态加密

- **定义：**
  - 同态加密（Homomorphic Encryption，HE）通过对相关密文进行有效操作（不需获知解密秘钥），从而允许在加密内容上进行特定代数运算的加密方法。



同态加密

- 传统加密方法要求在处理数据之前对数据进行解密，使其面临潜在风险。
- 同态加密是隐私计算的分支，允许在不泄露底层信息的情况下对加密数据进行计算。

- **特点：**
  - 允许在加密之后的密文上直接进行计算，且计算结果解密后和明文的计算结果一致。
- **意义：**
  - 同态加密的提出将加密技术研究从静态引向动态，是理论上的巨大革新，也开创了隐私计算的先河。

同态加密分类

- **全同态加密**（Fully Homomorphic Encryption, FHE）：算法支持对密文进行任意形式计算；
- **半同态加密**（Somewhat Homomorphic Encryption，SWHE）：支持对密文进行部分形式计算；
  - 如仅支持加法、仅支持乘法或支持有限次加法和乘法，则称其为部分同态加密 PHE（Partially Homomorphic Encryption）

- 一般而言，由于任意计算均可通过加法和乘法构造，若加密算法同时满足加法同态性和乘法同态性，则可称其满足全同态性。

同态加密分类

- 1978年，Rivest、Adleman（RSA 中 R 和 A）和 Dertouzos 提出全同态加密构想，自此成为密码学研究领域公开难题。

- **半同态加密算法：**主要包括以 RSA 算法和 ElGamal 算法为代表的乘法同态加密、以 Paillier 算法为代表的加法同态加密以及以 Boneh-Goh-Nissim 方案为代表的有限次数全同态加密；
- **全同态加密算法：**主要包括以 Gentry方案为代表的第一代方案、以 BGV 方案和 BFV 方案为代表的第二代方案、以 GSW 方案为代表的第三代方案以及支持浮点数近似计算 CKKS 方案等。

同态加密分类

类型		算法	时间	说明	实际应用
半同态加密	乘法同态	RSA 算法	1977	非随机化加密，具有乘法同态性的原始算法面临选择明文攻击	在非同态场景中应用广泛
		ElGamal 算法	1985	随机化加密	DSS 数字签名标准基于 ElGamal 数字签名算法的变体
	加法同态	Pailier 算法	1999	应用最为成熟	联邦学习
	有限次数全同态	Boneh-Goh-Nissim 方案	2005	仅支持1次乘法同态运算	/
全同态加密		Gentry 方案	2009	第一代全同态加密，性能较差	
		BGV 方案	2012	第二代全同态加密，性能相对较好	IBM HElib 开源库
		BFV 方案	2012	第二代全同态加密，与 BGV 类似	微软 SEAL 开源库
		GSW 方案	2013	第三代全同态加密，基于近似特征向量	TFHE 开源库
		CKKS 方案	2017	可实现浮点教近似计算，适合机器学习建模场景	HElib 和 SEAL

应用场景：云计算

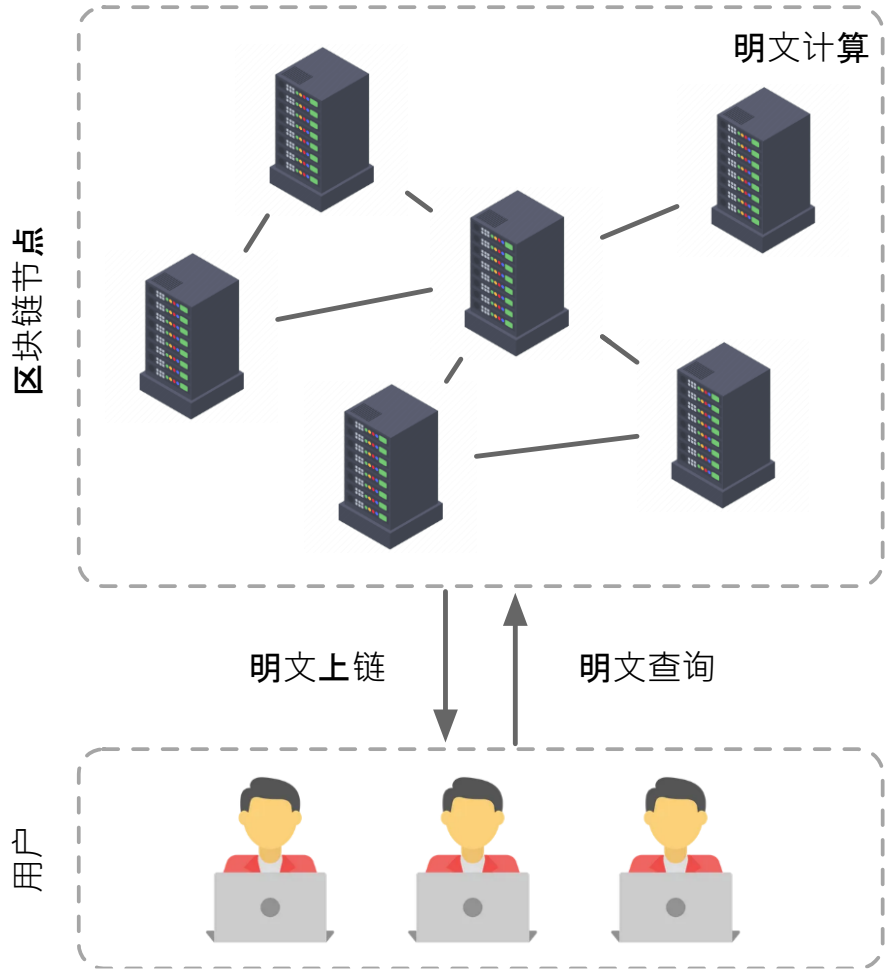
- 云计算中，利用云服务提供商强大的算力资源实现数据的托管存储和处理。但将明文数据直接交给云服务器具有安全风险，传统加密存储方式则无法实现对密文数据直接计算。
- 传统云存储与计算解决方案中，用户需要信任云服务提供商不会窃取甚至泄露用户数据，而基于同态加密的云计算模型可在根本上解决这一矛盾。
- 首先，用户使用同态加密算法和加密密钥对数据进行加密，并将密文发送给云服务器；云服务器在无法获知数据明文的情况下按照用户给定的程序对密文进行计算，并将密文计算结果返回给用户；用户使用同态加密算法和解密密钥对密文计算结果进行解密，所得结果与直接对明文进行相同计算的结果等价。

应用场景：区块链

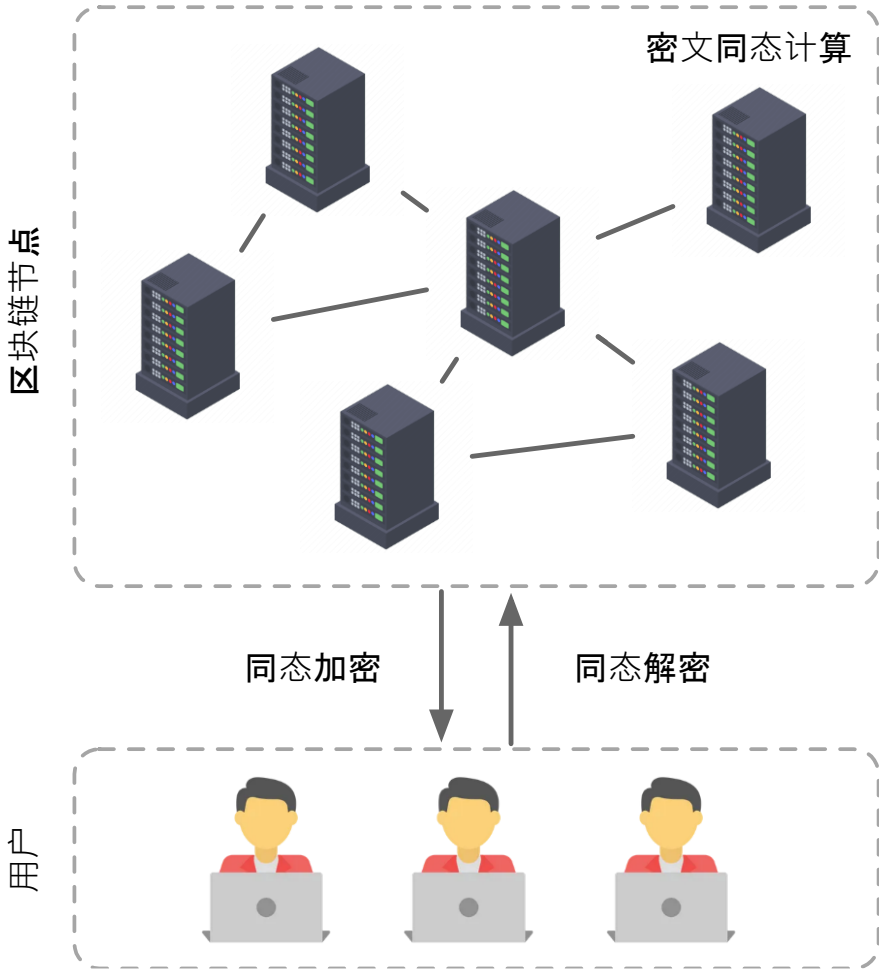
- 区块链应用基本逻辑是将需要存证信息上链，并通过众多区块链节点的验证和存储，确保上链数据的有效性和不可篡改性。
  - 例如比特币中，用户将转账信息进行广播，区块链节点在进行验证后将其打包上链，保证交易合法性；以太坊中，需要依赖区块链节点对智能合约正确执行，实现链上信息统一和正确。
- 但是，无论是公有链还是联盟链，直接基于明文信息进行区块链发布通常会在泄露一定的敏感数据。
- 对数据进行同态加密，并将计算过程转化为同态运算过程，节点可在无需获知明文数据情况下实现密文计算。例如，区块链底层应用平台特别是公有链平台大多基于交易模型，可考虑采用加法同态加密进行支持隐私保护的交易金额计算等操作。

应用场景：区块链

传统区块链应用模型



基于同态加密的区块链应用模型



同态加密优缺点

- **缺点：**
  - 同态加密的效率比较差，过高算力开销限制了其大规模应用，不适宜用于处理大规模数据，不适宜频繁变化的场景；
- **优点：**
  - 同态加密具有安全理论优势和多场景适配的优势。所以还需多挖掘应用场景，尤其是数据量小、变化小的场景。

细节 2：TEE 可信执行环境

可信执行环境 TEE

- **基本定义：**
  - 基于硬件隐私保护，计算平台上由软硬件方法构建安全区域，保证在安全区域内部加载代码和数据在机密性和完整性方面得到保护。
- **详细定义：**
  - 基于软硬件安全架构，通过复用 CPU 或划分部分内存为安全区域，构建出一个与外部相隔离的安全计算环境，所有敏感数据均汇聚在该安全区域内进行计算，未经授权访问 ，其他任何外部攻击者，包括系统管理人员均无法控制环境内的运算执行，也无法获取环境内的敏感数据，硬件隔离充分保证了环境内敏感数据 隐私计算的安全性。

TEE 历史

- **TEE 前身：**
  - 2009年，OMTP 工作组率先提出一种双系统解决方案：在同一个智能终端下，除多媒体 OS 外再提供一个隔离的安全OS，这一运行在隔离硬件之上的隔离安全 OS 用来专门处理敏感信息以保证信息安全。
- **实践层面：**
  - 隐私计算在可信执行环境（TEE）上相对成熟，其中 Intel 和 ARM 生态相对比较成熟。
  - 目前可信执行环境的实现技术主要有 Intel 的 SGX、ARM 的 TrustZone 等。

TEE 特征

1. **数据隔离：**
   - 可信应用的数据不能被其他应用访问、修改，包括可信应用数据对外 OS 隔离以及多个可信应用间数据隔离。
2. **计算隔离：**
   - 可信应用计算和资源不能被其他应用观测和拦截，同时需要清理可信应用执行后的痕迹，防止来自侧信道的攻击。
3. **通信控制：**
   - 非可信应用和可信应用、多个可信应用之间会话和数据交互不能破坏隔离性。
4. **错误隔离：**
   - 非可信区域的安全漏洞不能扩散到可信应用中。

TEE 生态

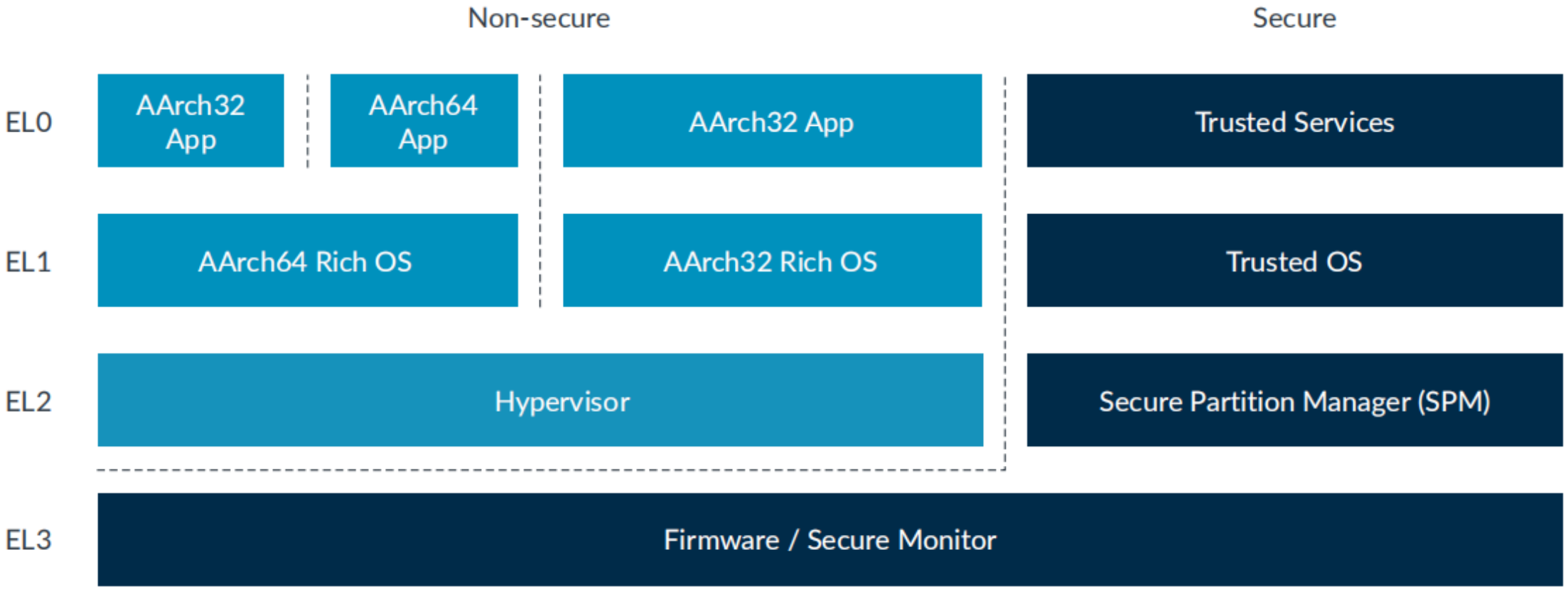
技术方案	国外					国内				
	Intel SGX	Intel TDX	ARM TrustZone	ARM CCA	AMD SEV-SNP	海光 CSV	飞腾 TrustZone	兆芯 TCT	鲲鹏 TrustZone	平头哥 玄铁
发布时间	2015	2020	2005	2021	2021	2020	2019	2017	2019	2021
指令集架构	X86_64	X86_64	ARM	ARM	X86_64	X86_64	ARM	X86_64	ARM	ANY
是否支持任意代码执行	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
硬件安全密钥	Y	Y	N	N	Y	Y	N	Y	N	Y
内存加密	Y	Y	N	N	Y	Y	N	Y	N	Y
内存完整性保证	Y	Y	N	N	Y	Y	N	Y	N	Y
TEE 安全I/0	N	Y	Y	Y	Y	Y	Y	Y	Y	Y
可用内存	<=1T	系统内存	系统内存	系统内存	系统内存	系统内存	系统内存	系统内存	系统内存	系统内存
TCB	• 硬件：CPU Package • 软件：Enclave内代码实现	• 硬件：CPU Package • 软件：intel Tox 镜像	• 硬件：安全虚拟核 • 软件：安全世界OS和TA	• 硬件：安全虚拟核 • 软件：Realm、安全世界OS和TA	• 硬件：AMD Secure Processo • 软件：虚拟机境像	• 硬件：海光SME • 软件：成拟机镜像	• 硬件：安全虚拟核 • 软件：安全世界OS和IA	• 硬件：CPU & TPCM	• 硬件：安全虚拟核 • 软件：安全世界 OS 和 TA	• 硬件：CPU &TPM

TEE 生态

- 目前以 Intel SGX 和 ARM TrustZone 为基础 TEE 起步较早，社区和生态已比较成熟。
- 国产芯片 TEE 方向上开始发力，如海光、鲲鹏、飞腾、兆芯等都推出支持 TEE 技术，信创国产化趋势明显，相关生态也正在加速建立、完善。

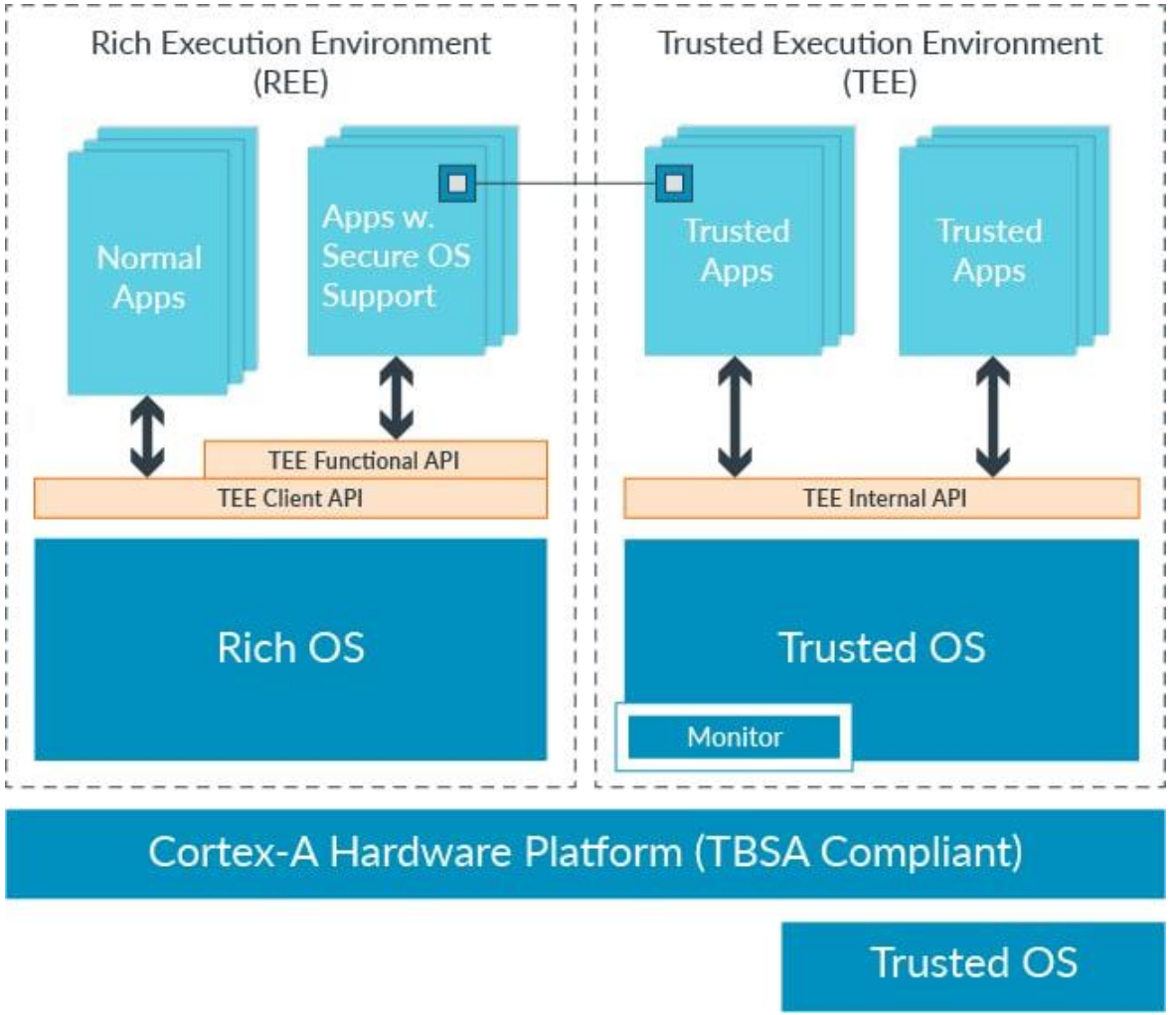
基于 ARM 架构 TEE

- **TrustZone**：ARM 提供的硬件级安全解决方案，旨在为处理器提供安全隔离和保护机制。将处理器内部划分为两个独立安全区域：安全世界和普通世界，从而实现了不同安全级别隔离。



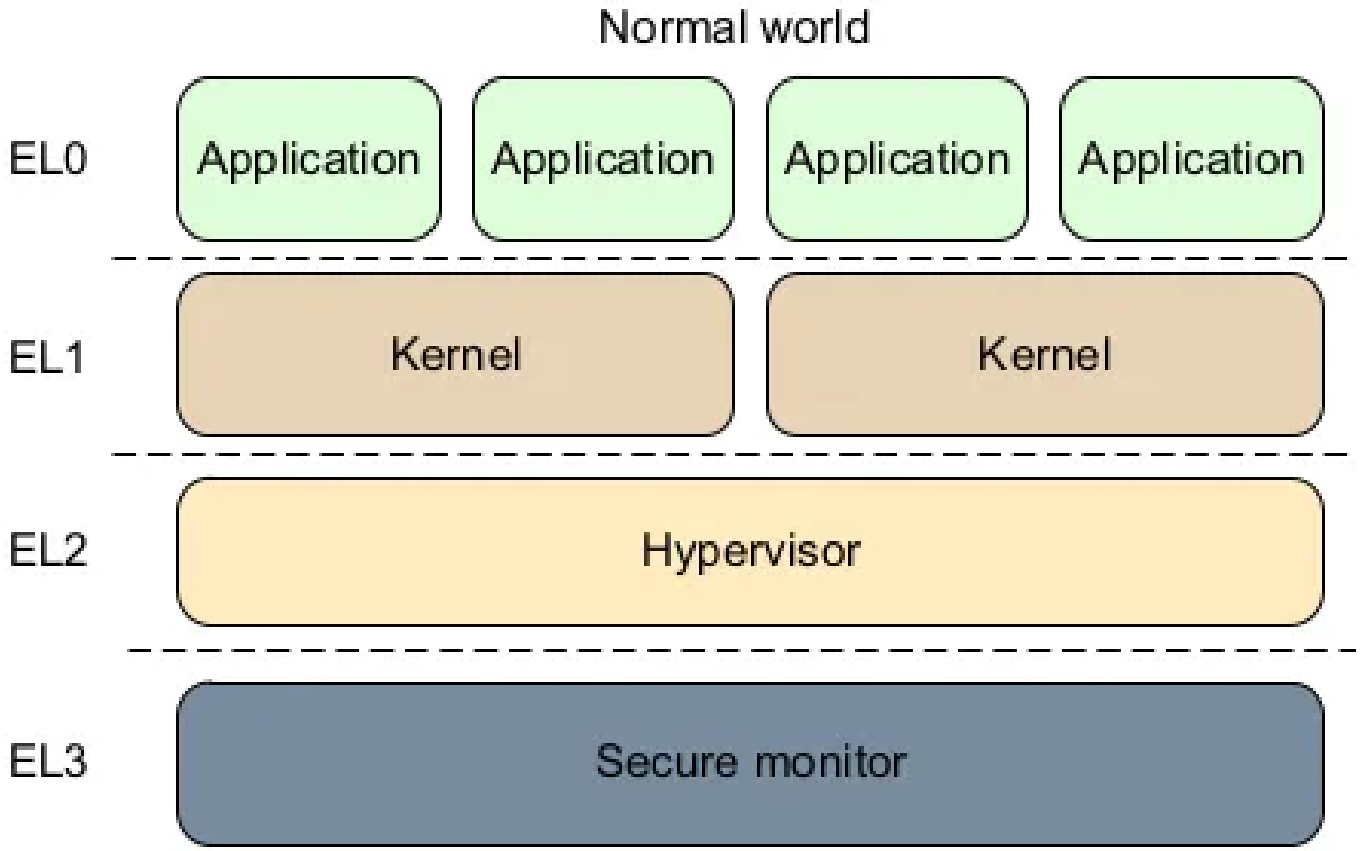
基于 ARM 架构 TEE：两种执行环境

- **正常世界：**运行一个丰富的软件栈，称为富执行环境，由完整的操作系统和应用程序组成， 它被通常认为是不可信任的；
- **安全世界：**运行一个较小软件栈，由可信操作系统和可 信应用组成。TrustZone 执行系统范围内的世界间隔离，并在富执行环境调用可信应用的服务时为世界切换提供受控的入口点。



基于 ARM 架构 TEE：异常级别

1. **EL0：Applications，**最低特权级别，在这个级别上执行的是用户应用程序代码。
2. **EL1：OS kernels，** 是操作系统内核运行的特权级别，也称为监管模式。
3. **EL2：Hypervisor，**用于虚拟化额外特权级别。
4. **El3：Secure monitor，**用于处理安全功能的额外特权级别。



可信执行环境

- 可信执行环境通用性好、准确性高，因原始数据不流出硬件隔离环境故安全性高，可单独用于隐私计算，也可以与其他隐私保护技术相结合，计算性能高，但需要确认硬件厂商是否可信。

细节 3： SMC 安全多方计算

安全多方计算发展历史

- 1981年，Rabin 首次提出通过 Oblivious Transfer(OT) 协议实现机密信息交互。
- 1982年，姚期智教授在论文《Protocols for Secure Computations》中提出“百万富翁问题“，即两个百万富翁在没有可信第三方、不透露自己财产状况的情况下，如何比较谁更富有。这标志着多方安全计算技术的产生。
- 1986年，姚期智教授提出混淆电路技术，实现了第一个多方（两方）安全计算方案。
- 1987年，Goldreich 等人提出基于电路的秘密共享方案GMW,并将其应用于多方安全计算。

- *中国计算机科学家、2000年图灵奖获得者姚启智教授*

安全多方计算

- 安全多方计算（SMC，Secure Multi-party Computation）将计算分布在多个参与方之间的密码学分支，参与者在不泄露各自隐私数据情况下，利用隐私数据参与保密计算，共同完成某项计算任务。
- 假设在一个互不信任的多用户网络中，n 个参与者 $P_1, P_2, \cdots, P_n$，每个持有秘密数据 x_i，希望共同计算出函数 $f(x_1, x_2, \cdots, x_n) = (y_1, y_2, \cdots, y_n)$，$P_i$ 仅得到结果 y_i，并且不泄露 x_i 给其他参与者。

安全多方计算：优缺点

- 安全多方计算因其基于密码学原理，安全性高，且不依赖于可信第三方，相对联邦学习和可信执行环境技术成熟度高；不过由于其基于密码学操作，随着参与方的不断增多会导致计算复杂度的增加。

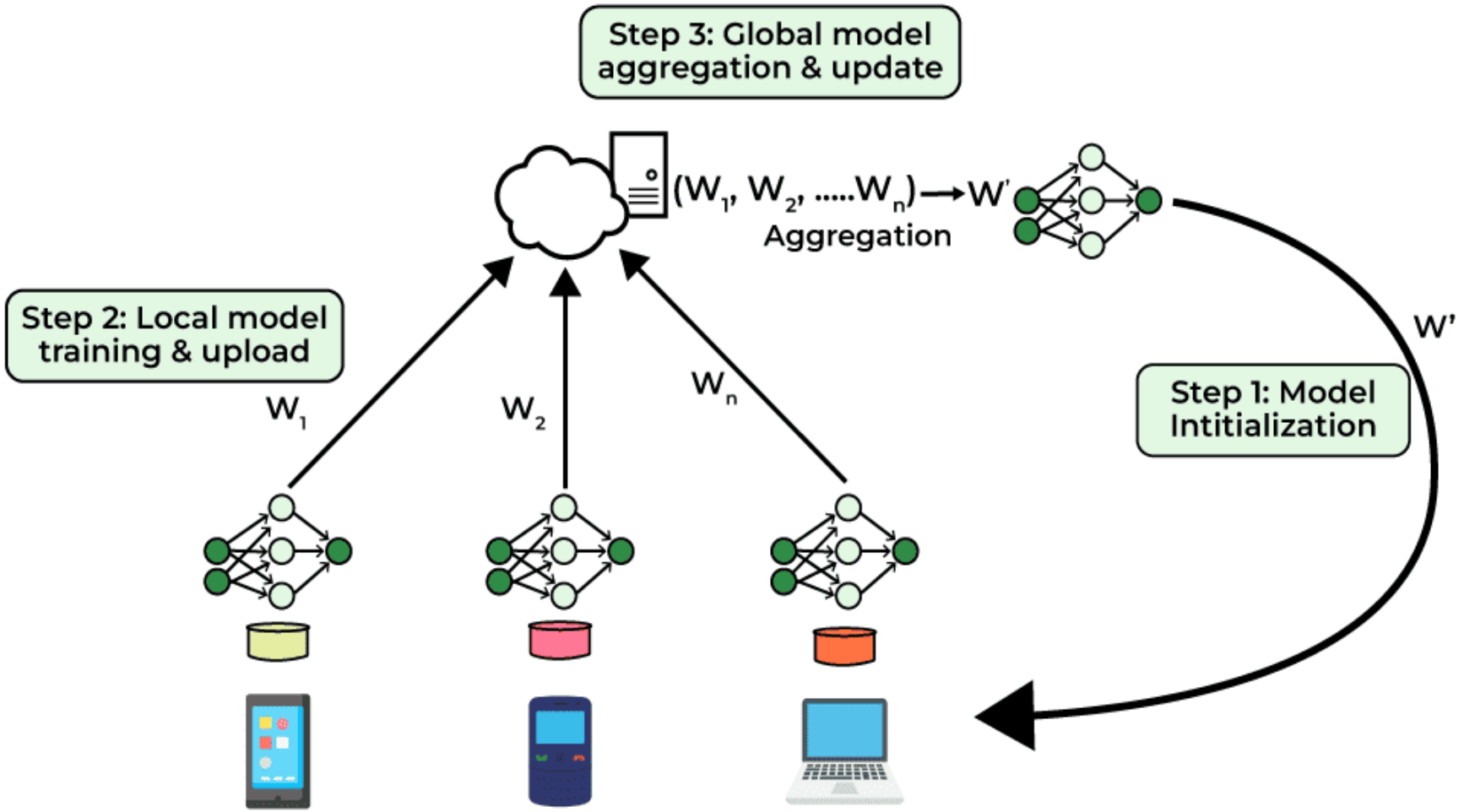
细节4：联邦学习

联邦学习

- **简单定义：**
  - 一种具有隐私保护属性的分布式机器学习技术。
- **详细介绍：**
  - 模型运算本地进行，只在各个参与方之间交换不包含隐私信息的中间运算结果，用于优化各个参与方相关的模型参数，最终产生联邦模型并将应用于推理。从而实现了“原始数据不出本地”、“数据可用不可见”的数据应用模式。

联邦学习流程

- 各参与方从中心服务器下载模型 A，然后使用本地数据进行训练
- 将训练后模型 A_i 加密上传至中心服务器
- 中心服务器将收集到各参与方模型 $[A_1, A_2, \cdots, A_n]$ 进行聚合计算
- 最终产生新的全局最佳模型 A^*。

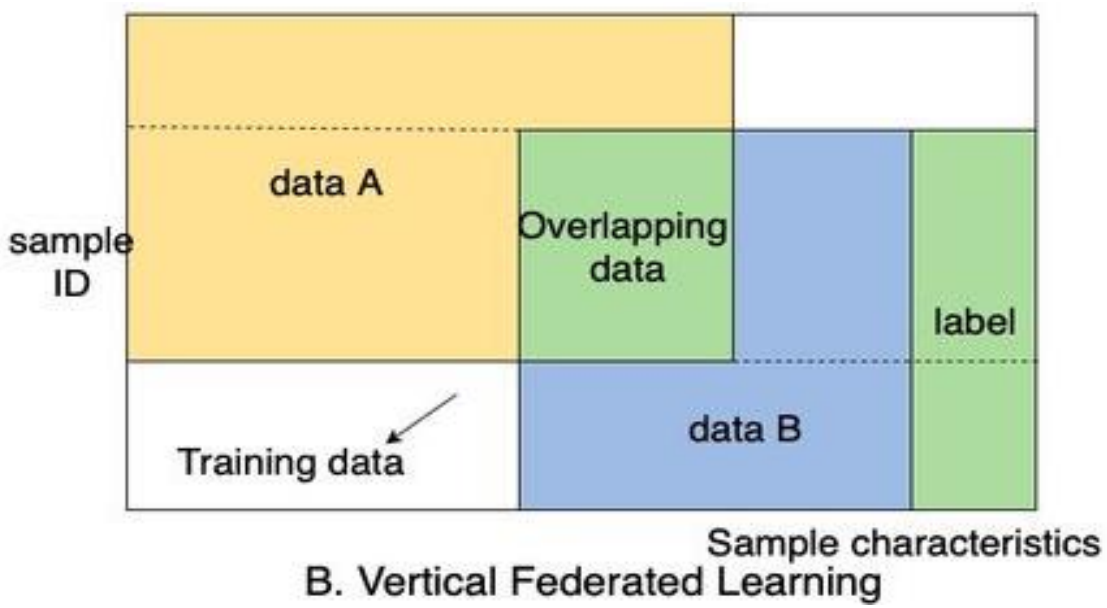


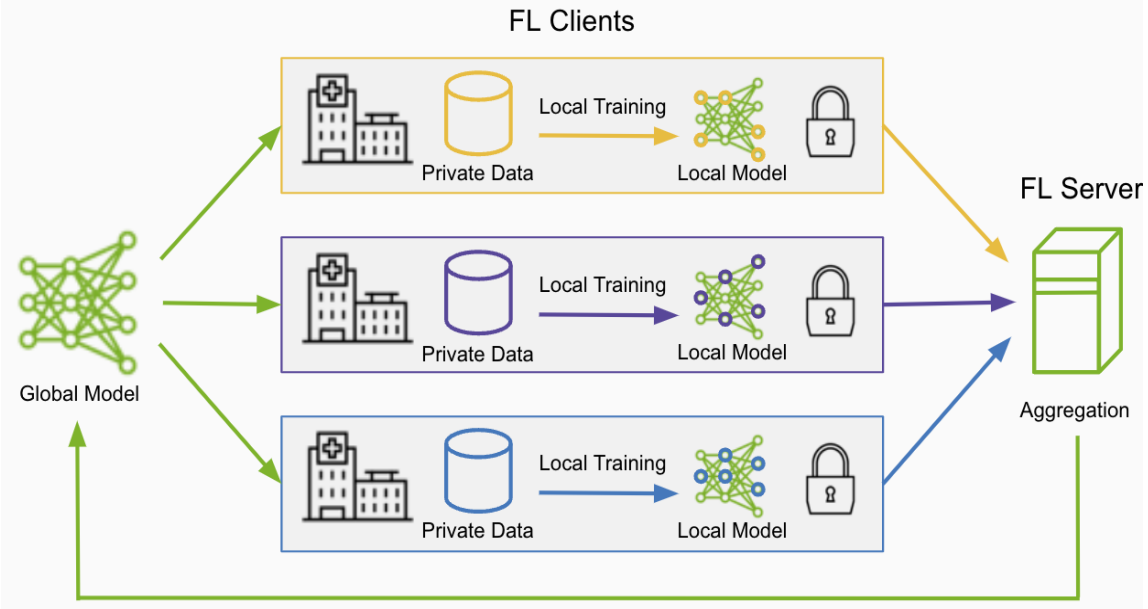
联邦学习分类

- **特点：**
  - 联邦学习以数据收集最小化为原则，参与方原始数据不出本地库，只交换加密的中间结果，有效实现参与方数据的隐私保护。
- **分类：**
  - 根据参与方提供的训练数据的样本和特征重合情况，分为横向联邦学习、纵向联邦学习以及联邦迁移学习。

横向联邦学习

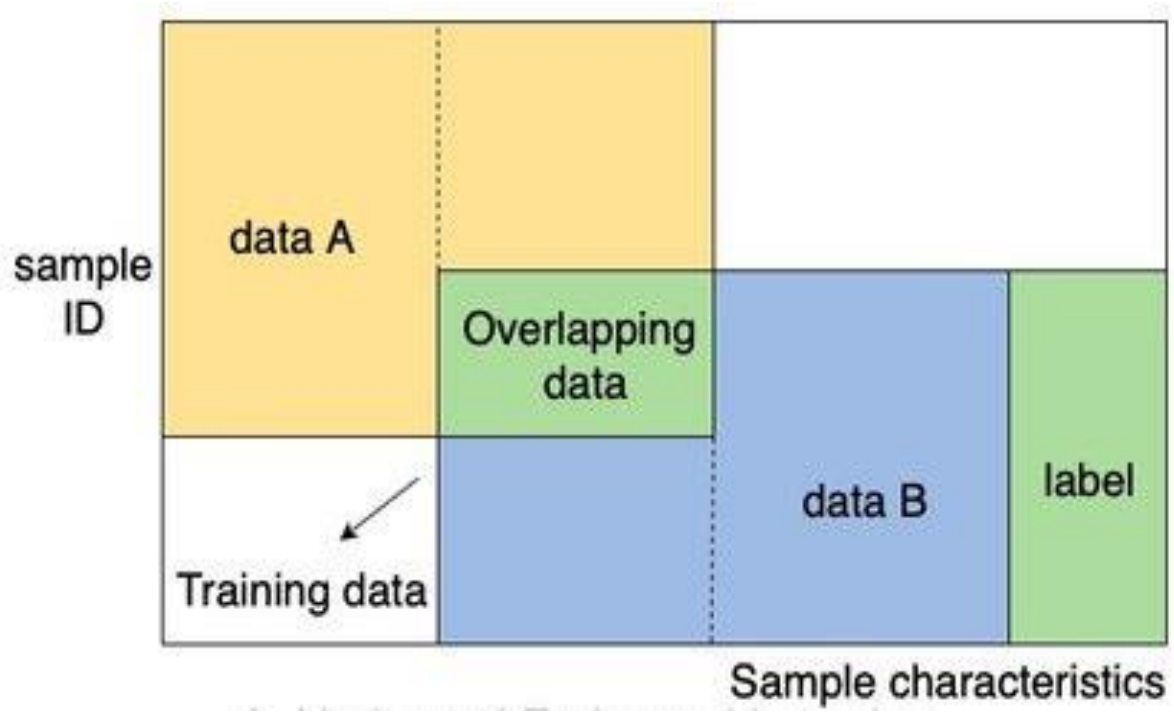
- 适用于样本重合度低、特征重合度高的场景，通过增加具有相同特征的样本数量来提升模型训练效果。

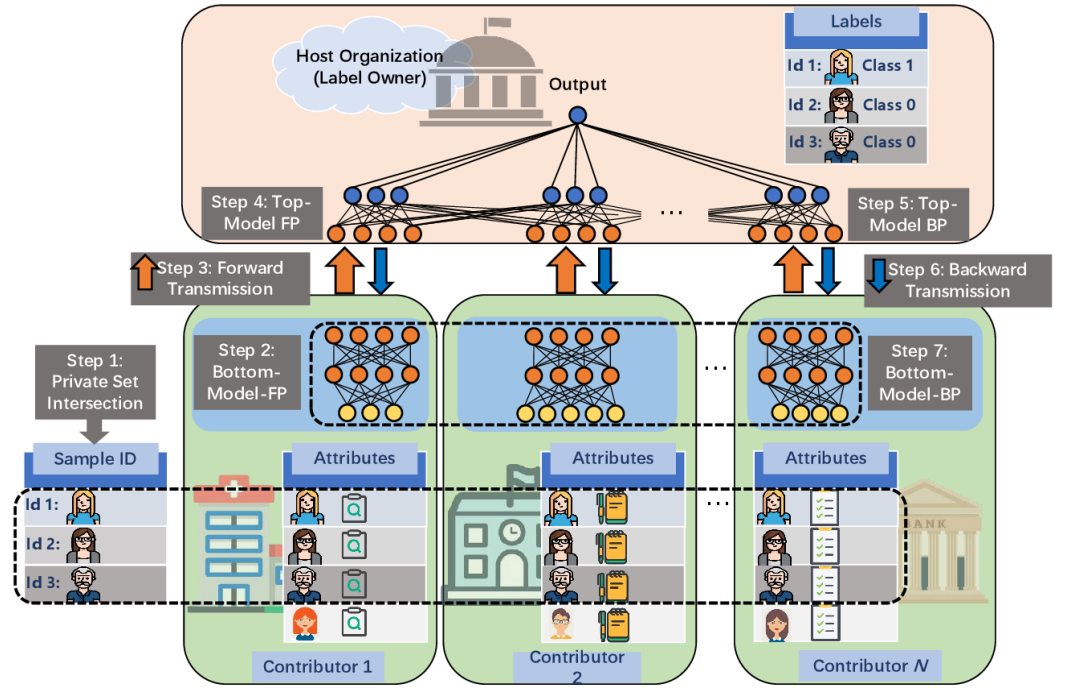




纵向联邦学习

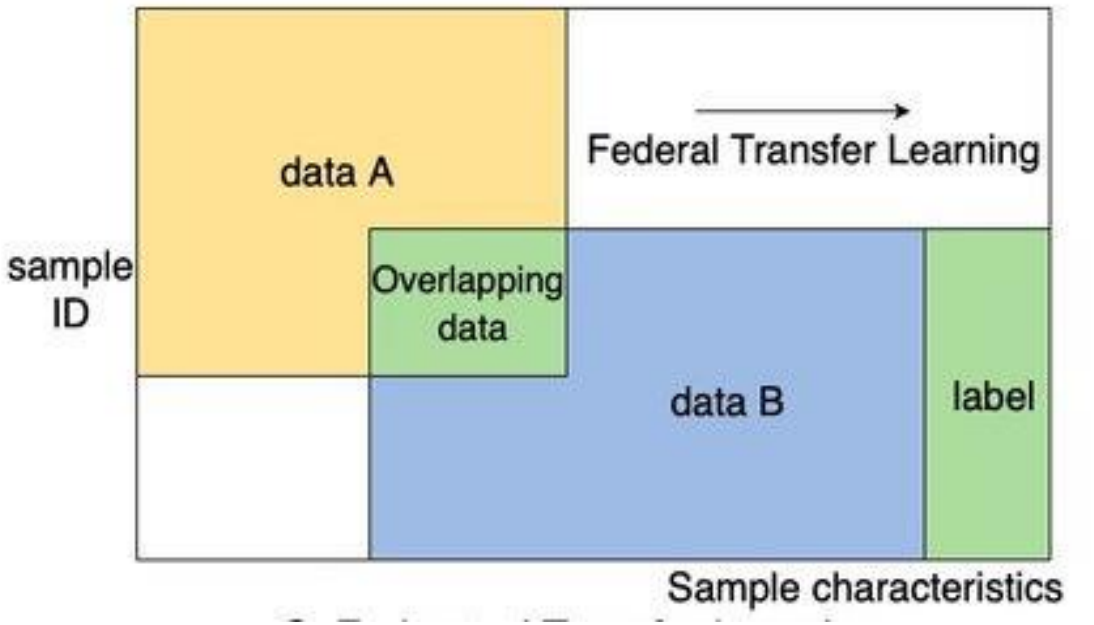
- 适用于样本重合度高、 特征重合度低的场景，通过丰富相同样本的特征维度来优化模型。

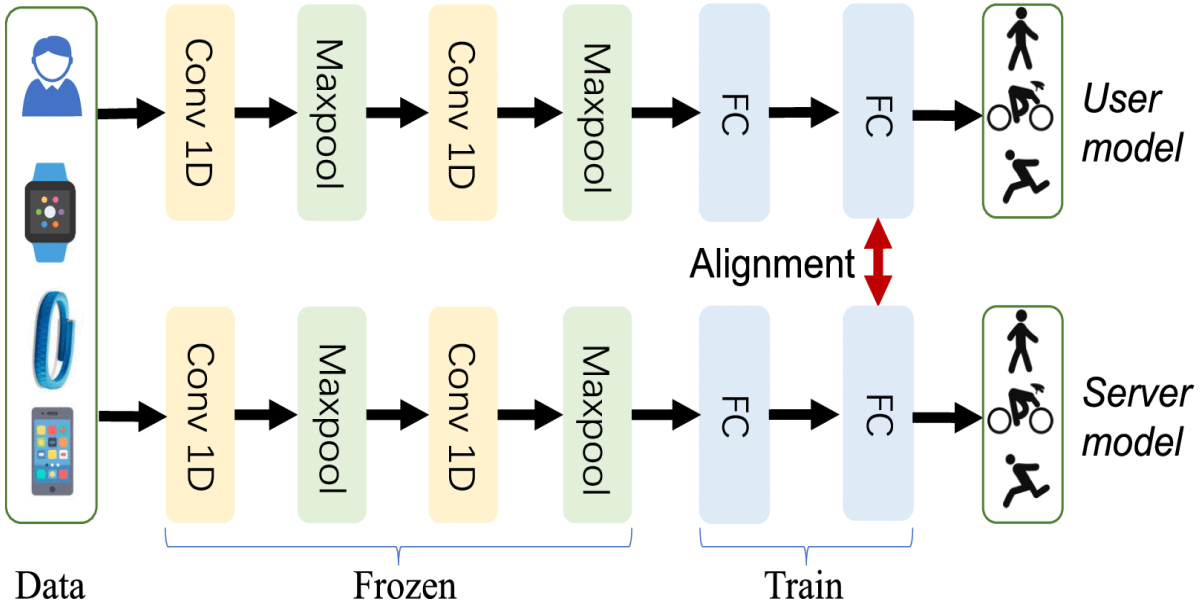




联邦迁移学习

- 适用于样本重合度低、特征重合度也低的场景，通过迁移学习解决单边数据规模小以及标签缺失或样本少的问题，以提升模型训练效果。





联邦学习：优缺点

- 联邦学习因原始数据不流出，一定程度上满足了数据的隐私安全需求，且不依赖于可信第三方，可解决算法复杂的建模问题，因此性能方面存在一定的瓶颈，需结合其他隐私保护技术才能保证数据隐私安全。

细节5： Others

零知识证明 Zero-Knowledge Proof, ZKP

- **历史：**
  - 1985年，S. Goldwasser、S. Micali和C. Rackoff首次提出零知识证明。实际应用中，某些加密货币就采用了这一技术路线。

- **定义：**
  - 证明者能够在不向监控者提供任何有用信息的情况下，使验证者相信某个论断是正确的。零知识证明实际上是一种涉及双方或更多方的协议，即双方或更多方完成一项任务需要采取一系列步骤，证明者需要向验证者证明并使其相信自己知道或拥有某一消息，但证明过程不向验证者泄露任何关于被证明消息的信息。

差分隐私 Differential Privacy, DP

- 2006年，C. Dwork 提出差分隐私，这一技术路线的主要原理是通过引入噪声对数据进行扰动，并要求输出结果对数据集中的任意一条记录的修改不敏感,使攻击者难以从建模过程中交换的统计信息或者建模的结果反推出敏感的样本信息。

Thank you

把AI系统带入每个开发者、每个家庭、每个组织，构建万物互联的智能世界

Bring AI System to every person, home and organization for a fully connected, intelligent world.

ZOMI

Course chenzomi12.github.io

GitHub github.com/chenzomi12/AIFoundation