December 2022

# ACPR Tech Sprint on Confidential Data Pooling

## Summary report

Author: Laurent Dupont, Fintech-Innovation Hub, ACPR

# 1. Executive summary

The Confidential Data Pooling Tech Sprint created and hosted by the ACPR in 2022 aimed to shed light on the various methods and techniques available for maintaining the confidentiality of sensitive data for collaborative analytics. It was a preliminary to the AML/CFT (Anti-Money Laundering/Combating the Financing of Terrorism) initiative launched by the ACPR in March 2022 and ongoing in 2023, of which it constituted a technological requirement.

The objective of the Tech Sprint was to **identify technological solutions enabling to ensure that multiple actors can securely collaborate using their respective sensitive datasets**. Within the more specific scope of the AML/CFT experimentation, relevant solutions were those supporting **the design and optimization of detection models (especially for suspicious transactions) operating on data pooled among peers, while guaranteeing that the data is processed in compliance** with any regulatory, legal, or organization-specific requirements.

An important feature of the challenge defined by the ACPR was that **proposed solutions could not be ranked**: rather than judging the absolute merits of each solution, the goal was to assess its technical capabilities and to empower each financial actor participating in the subsequent experimentation on sensitive data pooling to evaluate their adequacy to its own needs and constraints.

**Twelve solutions** were designed, implemented and presented for the Tech Sprint – some of them by a single startup and others by a team of well-established technology vendors. They **spanned a wide variety of technical and functional characteristics**: generic or specialized in AML/CFT; relying on an intermediary or fully decentralized; addressing input data privacy but also sometimes output privacy; with most of them offering various transaction monitoring capabilities.

Throughout this diversity, Tech Sprint solutions collectively reflected **the global state of the art in privacy-preserving technologies** and modern cryptography. Homomorphic encryption, secure multi-party computations, federated learning, secure hardware enclaves, and differential privacy were thus represented.

Among the key lessons learned from the initiative are the following:

- **All proposed methods proved quite amenable to the Tech Sprint challenge**, while each presenting their own strengths and limitations. For example, even the approaches considered *a priori* as the most resource-consuming (in terms of computational time or communication cost), such as the fully-decentralized ones or those based on secure hardware, were able to fulfill the "proof of concept" specified by the ACPR.
- The kind of data pooling mechanisms used were themselves quite diverse: basic or advanced linkage between transactional datasets, network construction, or simple on-the-fly unitary queries. **The data pooling mechanism turned out to be the most limiting factor among Tech Sprint solutions** due to the constraints it imposes on the types of detection models supported– and thus on the potential gains to be expected from data pooling.
- Besides already identified functional needs (e.g. that AML/CFT teams should retain ownership and control of the resulting detection models) and technical requirements (e.g. integration with financial institutions' IT), a dimension emphasized by financial actors invited to the Tech Sprint was the technical complexity of proposed solutions. Whether perceived or genuine, this complexity is a major hurdle to the adoption of the more sophisticated privacy-preserving technologies. On the long term however, those solutions offer undeniable benefits in terms of security guarantees, as brilliantly demonstrated by some Tech Sprint participants.

# 2. Introduction

## 2.1 Context and motivation for the Tech Sprint

The ACPR decided to launch in March 2022 an experimentation whose main objective is to prove – or disprove – the hypothesis that data pooling enhances the performance of transaction monitoring systems.

Questions raised by the various constraints associated to that primary objective led the ACPR to organize, as a preamble to the experimentation, an event called the Tech Sprint on Confidential Data Pooling (CDP). The Tech Sprint, which ran from May 2022 until the publication of this summary report, had itself a twofold objective:

- It primarily aimed to evaluate the various methods and techniques available for maintaining the confidentiality and integrity of pooled data.
- A secondary objective of the Tech Sprint was to elicit partnerships between financial institutions (FIs) participating in the experimentation - assembled in teams of two or more - and technology providers.

## 2.2 Timeline

The ACPR issued on May 16 a Call for Applications, and based on submissions received decided to select 12 participants in the Tech Sprint. Each participant consisted of a single technology provider or of several ones assembled as a team.

Each Tech Sprint participant was asked to implement a "Proof of Concept" (PoC) which was specified in the Requirements Document provided on June 13 by the ACPR. The deadline for the PoC completion was September 13, on the Tech Sprint Demo Day during which each participant was invited to present the results of their work. Additionally, a documentation of their work had to be delivered to the ACPR by September 3, and a deep-dive Q&A period extended following the Demo Day until September 23.

## 2.3 Object of this report

This report summarizes the variety of solutions presented in the Tech Sprint; it outlines the main lessons learned from the initiative; finally, each solution is factually and briefly described.

By necessity, this document contains a significant amount of technical vocabulary i

In order to meaningfully describe the main characteristics, similarities and differentiating aspects of CDP solutions, the content and the vocabulary of this report are somewhat technical. Readers are invited to refer to the glossary in Section 0 (or to any other source of documentation on privacy-enhancing or cryptographic technologies) for a definition of the most important technical terms.

# 3. Tech Sprint description

## 3.1 Confidential Data Pooling and relevant technologies

The design of the ACPR's 2022 Tech Sprint aimed to keep the spectrum of technologies assessed as broad as possible. An *ad hoc* phrase was therefore coined for the occasion: Confidential Data Pooling (CDP). Its definition is primarily functional rather than conceptual: it designates technologies that shall satisfy to the best extent possible the objectives of the forthcoming AML/CFT experimentation, namely guaranteeing that multiple parties can share their highly sensitive datasets while guaranteeing the security of that data. More precisely, CDP describes the set of technologies enabling to provide confidentiality (and ideally integrity) guarantees with respect to data used in the ACPR experimentation, from the data pooling stage, through detection model building and tuning, all the way to the evaluation of the gain in predictive performance.

The range of technologies analyzed thus has a significant overlap with Privacy-Enhancing Technologies (PETs), while covering an even broader scope.

## 3.2 Modalities

The ACPR provided Tech Sprint participants with a PoC scenario, described in the Tech Sprint PoC Requirements Document, which contained:

- a hyper-simplified use case (based on anticipated experimental protocols, with domain-specific functionality barely sketched);
- a link to fictitious datasets;
- a set of required or optional tasks to implement.

Each participant was required both to implement its solution to the PoC, and to prepare a demonstration of the benefits put forth for their solution (in the form of an algorithmic proof, architectural/design arguments, or any other form of security guarantee). The scope of the exercise aimed to cover the Tech Sprint scenario strictly speaking, as well as anticipate the application of any relevant solution to the following experimentation on real data – which aims at assessing the improvement of the predictive performance of AML/CFT detection models.

## 3.3 Guidelines

### Technological neutrality

As first guiding principle, the ACPR Tech Sprint was designed to be technologically neutral, with two corollaries:

- First, like all such initiatives around innovation in the financial sector led by the ACPR, the event was completely independent from the authority's supervisory role and segregated from its control missions.
- Further, the ACPR shall refrain both from promoting and from shedding negative light on any specific technological solution (even less on a specific technology provider) and neither encourages nor discourages its adoption by supervised entities in the financial sector.

As a consequence, neither the PoC requirements nor the conclusions from the CDP Tech Sprint should be interpretable as prescriptive with respect to financial institutions' technological choices. As explained in the previous section, the Tech Sprint was designed to consider the broadest spectrum of methods and algorithms possible. Section 4.1 clearly demonstrates that this objective was achieved.

### Definition of the Tech Sprint challenge

The principle of not excluding any confidentiality method nor any use case may be in tension with the ACPR's aim for its initiative to achieve concrete, tangible and quantifiable results. Therefore the Tech Sprint subject attempted to strike a balance between:

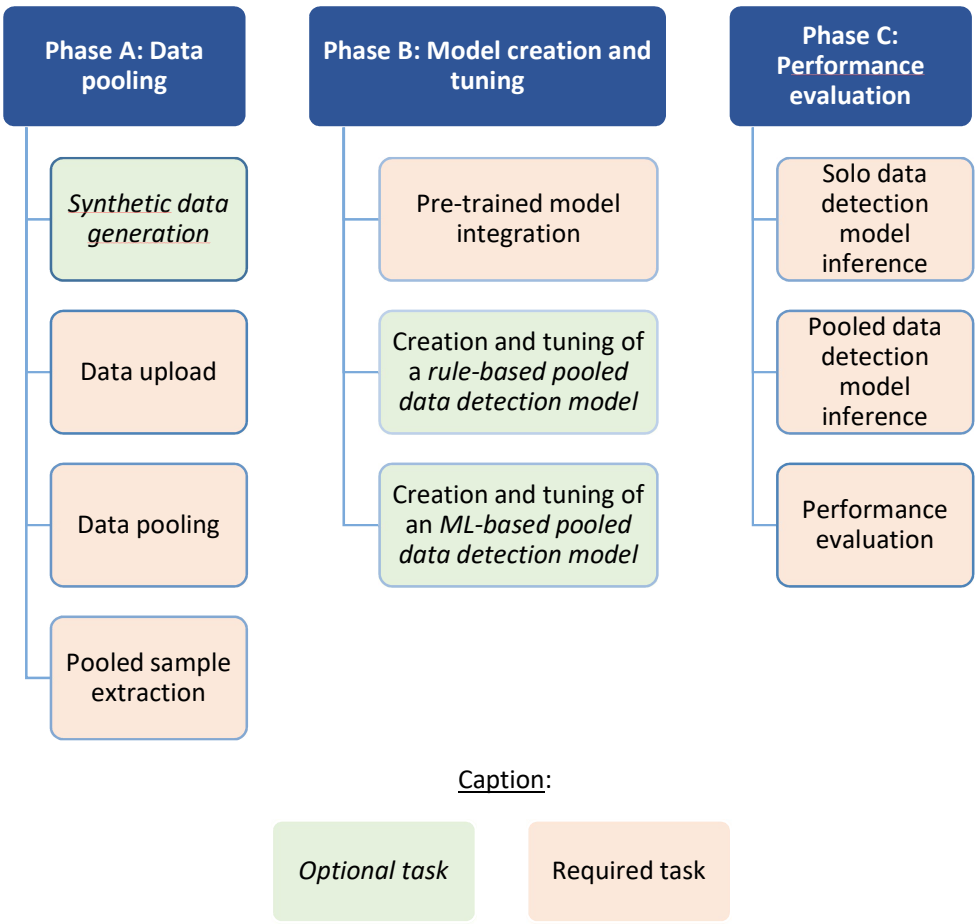
- on the one hand, the generality of technical requirements;
- on the other hand, a level of specificity and guidance which maximizes the likelihood of success in the following stages of the ACPR experimentation, i.e. the design and the execution of each experimental protocol on real data by participating FIs.

In practice, PoC requirements thus defined a list of features which mixed two types of functional requirements:

- must-have functionality expressed in terms as generic as possible
- nice-to-have functionality (such as ML models, integration of custom detection models, and DP).

## 3.4 PoC task assignment

The following diagram summarizes the list of tasks assigned to Tech Sprint participants by the PoC requirements document. Section 0 will describe how the 12 solutions handled each of these tasks.



### 3.5 Required documentation

Participants were required to produce a documentation of their solution, comprising the following elements for each of the PoC tasks:

- the level of confidentiality maintained for each type of variable, and how it is achieved;
- the threat model considered;
- what integrity guarantees are provided and how;
- performance (resources, runtime) on each target environment considered;
- whether and to what extent runtime performance is impacted by certain factors (e.g. real data vs. fictitious data).

### 3.6 Evaluation of Tech Sprint solutions

In practice, those works could not be evaluated like a typical computer science (or data science) challenge or hackathon. In contrast, neither the objective criteria of a typical hackathon (typically centered around the predictive performance of an ML model) nor the subjective criteria of the 2021 ACPR Tech Sprint on AI explainability (scientific innovation, clarification of regulatory issues, etc.) were suitable in this case.

The main reason is that the challenge proposed did not lend itself to a one-size-fits-all solution: some solutions will be best suited for moderate amounts of sensitive data but not scale as well to massive datasets, while other solutions propose looser confidentiality guarantees but better scalability. Conversely, each FI has its own objectives and constraints, which prevents from defining simple ranking criteria that would coincide with the partnership decisions ultimately made by FIs participating in the AML/CFT experimentation: the Tech Sprint evaluation was not about assessing each solution's absolute merits, but its merits relative to each participating FI's needs and constraints. Further, the FIs being the sole responsible for those technical partnerships, the Tech Sprint did not aim to predetermine their decision.

Therefore, the main goal of the assessment of works performed and of deliverables produced was to give each participating FI the best information possible to choose a provider for the subsequent AML/CFT experimentation on real data. To this aim, the ACPR had assembled a panel of technical experts, whose role was to inform and guide the choice of a solution by each participating FI. The panel of technical experts for the Tech Sprint was composed of 1 to 3 members representing each of the FIs participating in the AML/CFT experimentation, 2 ACPR representatives (1 Computer Scientist + 1 Data Scientist) and 2 Banque de France representatives (1 Data Scientist + 1 IT Expert). Each member of the panel was selected for their expertise in data security, i.e. in cryptography (ideally including PET) and/or confidential data science, with a bonus for a cybersecurity expertise.

#### Input for the evaluation process

The information used for evaluating Tech Sprint solutions should take into account the entirety of materials provided to the panel of technical experts, namely:

- responses to the Call for Applications received from each candidate provider (between May 16 and June 13);
- the documentation of each solution (sent by September 3 and analyzed before September 13);
- the presentation of each solution on the Tech Sprint Demo Day (on September 13);
- the answers to any deep-dive questions asked following each presentation (Q&A period from September 14 to September 23).

## Evaluation form and bespoke criteria

Prior to the Tech Sprint Demo Day, the panel had met to discuss evaluation principles and criteria, enabling them to converge on an evaluation form, whose template is given as appendix (Section 7).

This evaluation form (and the criteria within it) was provided to participating FIs as mere guidance and not as prescriptive evaluation principles. Some FIs thus opted to use other criteria than those suggested by the ACPR, or to put a different emphasis on those criteria – which was also a positive sign of participating FIs' involvement and active commitment in the ACPR experimentation.

A few bespoke criteria put forward by participating FIs were the following.

### *Model ownership*

One FI emphasized their need to retain full ownership of the resulting detection models, which is a logical expectation but would need to be explicitly guaranteed from any solution deployed in a real-world scenario.

### *Technical complexity*

The level of complexity (otherwise put, of sophistication) inherent to the security capability of each solution was also mentioned by some participating FIs as an important evaluation criterion: assuming some (if not all) solutions would offer an equivalent level of confidentiality throughout the process, this level of complexity might turn out to be a decisive factor of selection (to be minimized of course).

### *Integration issues*

Some participating FIs naturally had their own integration constraints, which in some cases led them away from hardware-based solutions such as TEEs (Trusted Execution Environments) due to cost and IT policy considerations.

# 4. Overview of Tech Sprint solutions

## 4.1 Diversity of works

Solutions designed, built and presented for the Tech Sprint spanned a wide variety of techniques and algorithms, while also representing the state of the art in modern cryptography and privacy-preserving technology. Their main characteristics may be summarized in all their diversity as follows.

***Technologies***
They cover a wide range of privacy-preserving technologies (see next section).

***Consortiums vs. single providers***
Some solutions were proposed, built and presented by a group of collaborating companies (in some cases this partnership pre-dated the Tech Sprint, in others it was assembled specifically for the event). Others were all-in-one solutions, in general .already developed by a single provider (often a startup) and tailored to the challenge.

***General- vs. special-purpose***
Some solutions are general-purpose collaborative analytics platforms, whereas others are specialized AML or anti-fraud solutions.

***Hosting and deployment***
They support various hosting and deployment approaches: on-premise deployment, hosting on a public cloud (often with major security guarantees), or in some cases without requiring any deployment.

***Decentralized vs. intermediated***
Some solutions necessitate the designation and involvement of a trusted third party, while others are fully decentralized.

***Transaction monitoring capabilities***
Solutions offer various transaction monitoring capabilities: general-purpose supervised or unsupervised ML, graph-based analytics, or use case-specific capabilities.

***Flexibility and ease of use***
Many solutions offer a modular, extensible architecture, with most of them taking a developer-friendly approach by exposing an API[1], an SDK[2] and/or a DSL[3].

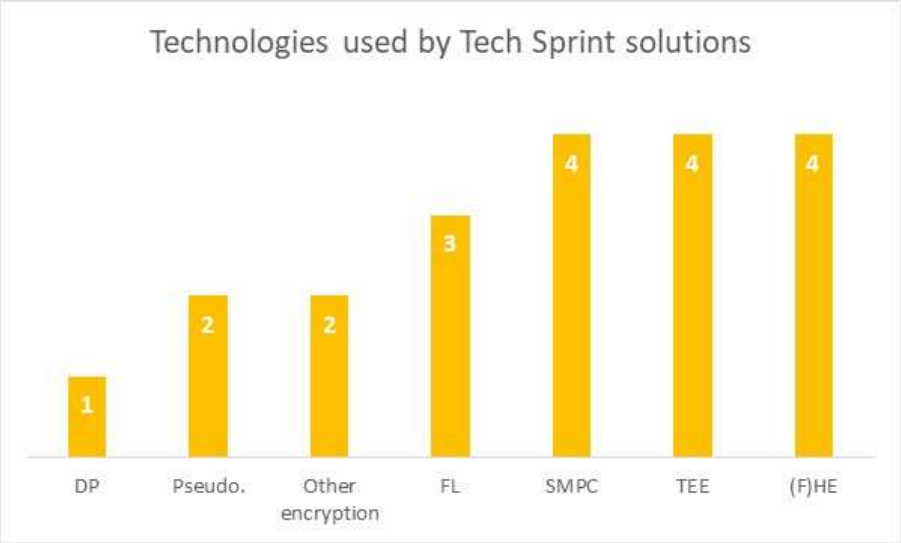
[1] API (Application Programming Interface) refers to the accessibility of external code through a simplified interface which enables extending a set of software features.

[2] SDK (Software Development Kit) refers to an entire kit to facilitate access to one or several APIs, and providing a set of software libraries, documentation, examples, etc. in one or more programming languages.

[3] DSL (Domain-Specific Language) designates a more or less rich and complex programming language, specifically created to communicate with a piece of software or control its functionality, so as to produce code that is both more concise and more expressive than by using a standard programming language (as in the case of an API or SDK).

### 4.2 Technologies used

The range of technological solutions, methods and algorithms proposed by participants can be broken down according to the type of privacy-preserving technology used (several of them being combined in some cases). The following chart indicates the number of Tech Sprint solutions relying on each of the main types of privacy-preserving technology, whether for preserving input privacy (and in some cases for providing integrity guarantees) or output privacy.




*Abbreviations used in this chart*

**DP**: Differential Privacy
**(F)HE**: (Fully) Homomorphic Encryption
**SMPC**: Secure Multi-Party Computation
**FL**: Federated Learning
**KMS**: Key Management System
**TEE**: Trusted Execution Environment

It should emphasized that these technologies are often used in conjunction with one another. In particular, TEEs are often deployed together with either full encryption methods (associated to a KMS, see next paragraph) or to a mechanism performing pseudonymization of sensitive attributes.

Confidentiality guarantees (and in some cases integrity guarantees) associated to each type of privacy-preserving technology used in the Tech Sprint are summarized in what follows.

#### Secure enclaves (TEEs)

TEE-based solutions usually assume that a remote attacker has gained full remote access to the CDP platform, including the infrastructure and other tenants of the Confidentiality Perimeter and of the enclave itself. Security properties of the TEE technology ensure:

- that all computations (data pooling, model creation, tuning and inference, results evaluation) are executed on a secure hardware enclave (by relying on a remote attestation process and guaranteeing that the attestation is actually signed by an enclave such as SGX and not by some simulator);
- that no party, including the enclave administrator and operator, has tampered with computation code or parameters;
- that the data and the code can be read solely by the secure enclave and not by its administrator or operator;
- that the list of the identities of each party involved in the computation is as expected;

- that the entire software stack used in computations is correctly identified and fully verifiable.

### Homomorphic encryption (HE, FHE)

HE-based solutions assume an "honest-but-curious" adversary (see glossary). The type of encryption used (which is probabilistic by nature) ensures that every time the data is encrypted, the encrypted representation will appear different from prior encryptions to a potential adversary. This property prevents man-in-the-middle attacks (i.e. in the Tech Sprint scenario, prevents a malicious party from corrupting the result that an AML/CFT analyst would receive).

HE-based aggregation is thus resistant to such attacks, however model privacy issues remain at the output, furthermore FHE requires the absence of collusion among participating nodes.

### Secure multi-party computation (SMPC)

Most SMPC-based solutions also assume an honest-but-curious adversary. Within this category , two approaches can be distinguished:

- non-verifiable secret sharing (such as Shamir secret sharing), which does not have a way to verify the correctness of each share during the share reassembly process;
- and verifiable secret sharing, which in theory can verify that shareholders are honest and not submitting "fake" shares.

### Differential privacy (DP)

DP-based solutions provide output privacy guarantees: the threat model in that case pertains to people or machines executing queries via an API, i.e. data scientists in the data exploration or model building stage, developers working on system integration, or IT staff working on deployed models. In those 3 situations, output privacy solutions reduce the risk associated to the leakage of sensitive data since any outgoing information would conform to the input data owner's confidentiality policy.

Overall, solutions without any output privacy are less suited to the proposed scenario. DP – or a similar method for reducing the re-identification risk – has a clear twofold benefit:

- The main advantage of DP in general is to limit the impact of any potential data leak
- An additional benefit, which is specific to the ACPR experimentation on AML/CFT, is that DP facilitates the specification and the implementation of a manual review process of alerts resulting from a detection model's inference on pooled data.

These two benefits are cumulative with other privacy-preserving methods since DP can be implemented on top of any solution providing input privacy.

### Federated Learning (FL)

Although 3 of the Tech Sprint solutions provide support for federated learning on confidential data, only one actually relied on FL to implement the PoC scenario. This confirms a prior hypothesis that, even though FL is at first glance appealing for this kind of scenario because it is (with some SMPC variants) one of the few methods which do not require any data leaving its owner and thus provides some of the strongest confidentiality guarantees, it is not quite appropriate to an AML/CFT scenario as it imposes too strong assumptions on the modelling process. Its main drawback is that the FL aggregation process must treat identically the model updates from each of the participants, and thus only supports the stacking setup described in Section 0 (and not the standard or custom join setups).

### Pseudonymization

The re-identification risk can also be reduced by more traditional means than DP, e.g. pseudonymization which in principle eliminates the risk of accidental or deliberate re-identification in cases where the platform leaks some data.

### Other cryptographic approaches

Besides the aforementioned methods based on innovative – and often somewhat complex – technologies, other approaches relied on more traditional end-to-end encryption combined with the use of a trusted third-party.

Some solutions also relied on a public cloud which also operates its KMS, so that the cloud operator and administrator all need to be trusted to not access any data in the clear nor to tamper with the executable code.

### Security of data in transit

In addition to all of the abovementioned threat models and associated security guarantees applying to data in use, one proprietary, highly innovative Tech Sprint solution provides a very strong threat model for data in transit. Although this kind of security guarantee fell outside the scope required by the Tech Sprint, this innovative and robust approach warranted a brief description:

- It builds a "secure overlay network" ensuring perfect secrecy at the node level (IP and path are concealed from participating nodes and from servers)
- Its ring architecture provides security and integrity guarantees as well as proof of origin (i.e. that data comes from a ring member)
- It is also resistant to MITM attacks thanks to distributed exit nodes, thus requiring no trusted third party at the network level.

## 4.3 General characteristics

This section presents general characteristics of the Tech Sprint solutions. Because those solutions cannot be meaningfully ranked based on intrinsic or absolute quality, highlighting their similarities and divergences should help understand their respective strengths and weaknesses, and the particular settings and scenarios for which they are best suited.

### Scalability and runtime performance

***The challenge of comparing runtime performance***

The Tech Sprint PoC requirements document asked for execution time measurements in order to broadly assess the comparative speed and scalability of the proposed solutions. Such comparative analysis proved however inherently difficult to perform since execution times largely depend on hardware characteristics and on the exact definition of measured tasks, which are often not mutually comparable themselves. For example, in some SMPC-based solutions data remains in its original location: the data upload task is virtual and its execution time cannot be measured, in contrast with solutions that perform a physical, time-bounded data pooling step.

More precisely, runtime performance indeed depends on many factors, including the type, size and number of datasets, the type of pooling and computation performed, the characteristics of the machines used for computations, the level of security that is required by encryption schemes (symmetric, asymmetric and homomorphic encryption), and more. Even further, the logic for data pooling and matching is ultimately defined by the end users (financial institutions in the case of the

ACPR experimentation), and - as in standard computations - the efficiency of the scripting logic and the amount of allocated computing power are the main driver of the time to compute.

In addition, only a minority of Tech Sprint participants reported the runtime for completing each PoC task by their solution.

For these two reasons, the documentation provided by Tech Sprint participants did not suffice to perform a comprehensive comparison of the scalability and runtime performance of each solution.

***General observations***

However some general tendencies regarding scalability can be drawn from the analysis of Tech Sprint results. In particular, two PET categories (SMPC and TEEs) are often considered as very prohibitive in terms of resources dedicated to communication between nodes or computation within a node:

- SMPC typically induces a large communication overhead, which in the case of the Tech Sprint may lead to strong restrictions on the classes of detection models that can be trained with privacy guarantees. SMPC algorithms tend to present some general characteristics: internet connectivity will negatively affect the algorithm's runtime, and performance generally increases as more participants join the virtual computational pool (as opposed to computations done in an environment where data is physically pooled).
- TEEs typically induce a large computational overhead, so that training a model inside an enclave is often quite slow[4] (among other reasons because it runs in a single-threaded process). A rough estimate of the overhead due to the in-memory encryption and decryption inside a TEE is that most scripts take 5-15% longer than the same script using the same amount of resources in a standard computing environment, with the worst-case performance experienced on a real-world query against real data being typically a 50%-100% overhead.

Interestingly, even the approaches considered a priori as the most resource-consuming, such as SMPC as TEEs, proved quite amenable to the PoC scenario. Indeed, their execution times on standard servers or VMs (Virtual Machines) turned out – when they were reported – to be comparable with other solutions or at most 1-2 orders of magnitude higher (but justifiably so due to the stronger security guarantees provided).

## Traceability and auditability

The PoC requirements document presented traceability and auditability requirements as an intrinsic part of desired solutions. In particular, mandatory task 4 consisted of a sample extraction capability against the results of data pooling, and mandatory task 10a consisted of a sample extraction capability against the results of PDDM inference.

Many solutions proposed for the PoC either neglected these traceability and auditability requirements, or simply reduced them to making application logs accessible but without any concern for the sensitivity of the underlying information. Even the latter option is an issue: if logs contain granular input or intermediary processing data they might lead to a confidentiality breach, conversely if they simply omit detailed information in order to preserve sensitive data they be of limited practical value for debugging.

By contrast, one TEE-based solution stood out by offering:

[4] In a near future however, the performance of TEEs is expected to greatly improve, especially as a new generation of confidential computing on GPU will offer much faster model training (e.g. the Nvidia H100 processor).

– Complete audit logs generated within the confidential computing environment. These logs allow full auditability and transparency on the actions performed in the Confidentiality Perimeter (making it possible e.g. to review when the computation has been run within the enclave and by which application or user).
– Error messages reported at runtime directly to the user. Those messages contained the same level of information as the full traceback would have in the clear, while also preserving privacy by redacting sensitive information.

Some solutions also have intermediate capability in this regard: they produce a log of all actions performed by application users, but not to the extent of the previous solution which provides a genuine data science workbench operating on confidential data.

### Cryptographic key management

Most Tech Sprint solutions rely on a KMS to store and give access to encryption keys.

For TEE-based solutions, secret keys are sealed automatically in the secure enclave, and are thus not visible by anyone, even to the participants.

Some solutions provide a BYOK (Bring Your Own Key) feature, whereby participants can provide their own, pre-existing key rather than having the system generate a new key or key pair.

Also, some solutions provide compatibility with third-party key management services (e.g. Microsoft Azure Key Vault and its managed Hardware Security Module capability).

Finally, other solutions do not address encryption key commissioning at all: all participants are assumed to share a key ring through a secure key distribution protocol and to handle key generation on their own.

### Sensitive data processing

The PoC requirements classified input variables in three categories:

- non-sensitive variables (visible during manual review);
- predictive variables (with predictive power for the detection models, not to be displayed during manual review);
- joining variables (useful for joining datasets, but with no predictive power or voluntarily excluded from the set of predictive variables).

Many solutions proposed by participants treated these 3 categories as required by the challenge specifications, nevertheless a few solutions did not accommodate for the second category, namely predictive variables, insofar as those solutions removed all sensitive attributes following the data pooling operation. This is an important limitation in terms of generic features, and for the purpose of the ACPR experimentation prevents training predictive models on confidential transactional attributes or on sensitive customer information (i.e. attributes which have predictive power but should not be visible by a collaborating FI).

## Confidential modelling capability on pooled data

A fundamental characteristic of Tech Sprint solutions is the extent of their support for creating and optimizing detection models operating on pooled data, while of course providing confidentiality guarantees during these creation and optimization stages. In other words, applying CDP to an AML/CFT modelling scenario precisely means providing such a capability. For example in the case of ML-based detection models, an adequate solution should enable data scientists to setup and train an ML model using pooled data containing sensitive attributes just (or almost) as if they were following a standard workflow on non-sensitive data[5].

A key insight from the Tech Sprint is that a solution's confidential modelling capability on pooled data directly derives from the implemented data pooling mechanism. In practice, the type of data pooling mechanism was found to be the most limiting design factor among Tech Sprint solutions due to the constraints it imposes on the types of detection models supported – and thus on the potential gains to be expected from data pooling.

The following tables describe the main types of confidential modelling capability on pooled data presented in the Tech Sprint, along with their respective benefits and limitations, and makes this relationship between CDP and confidential modelling explicit. They also display the number of solutions corresponding to each type of modelling capability[6].

| **1. On-demand querying** | Supported by **2** solutions |
|---|---|
| **Definition** | |
| An on-demand querying capability means that a solution provides a runtime query capability on specific data items (in the case of the PoC, typically a query to retrieve the transactional attributes of a customer or a set of customers). | |
| **Benefits and limitations** | |
| This capability is far more restrictive than support for creating and optimizing a detection model: for one thing, the latter enables tuning an entire model (whether rule-based or ML-based) on historical data, whereas on-demand queries require runtime access to the underlying pooled data store for their execution.<br>It should be noted however that a few Tech Sprint solutions offered on-demand querying *in conjunction* with model building and tuning on pooled data. | |
| **Detection models supported** | |
| None, only runtime querying is possible. | |

| **2. Stacked data modelling** | Supported by **5** solutions |
|---|---|
| **Definition** | |
| This type of modelling capability derives from pooling mechanisms which:<br>- require all datasets to share the same input schema (or a common subset of input fields)<br>- proceed by simply stacking those datasets (i.e. in the case of relational databases, by performing a UNION operation). | |
| **Benefits and limitations** | |
| Dataset stacking was widely adopted among Tech Sprint solutions as it provides simplifying assumptions for the overall solution (essentially, pooled data can be treated identically to solo data). | |

[5] Similar functionality was expected in the case of rule-based models, although the bar is then much lower since rules can be predefined without needing to consider training data. Thus the confidentiality level of input data has far less impact on the modelling workflow than in the ML case.

[6] The total number of occurrences adds up to more than 12 due to some solutions having dual capability, e.g. on-demand querying and modelling on joined data.

| |
|---|
| Conversely, the type of detection models supported is limited compared to joined data modelling, because dataset stacking treats each participating FI identically. Specifically, for a feature X present in datasets from A and B, such a model will take as input a single feature X defined on the stacked datasets. |
| **Detection models supported** |
| Stacked data modelling enables creating and tuning PDDMs which:<br>- use either the common data schema or the common subset of input fields as input features<br>- can be trained on the union of input training datasets (which is called horizontal federated learning in the case of FL). |

| **3. Joined data modelling** | Supported by **5** solutions |
|---|---|
| **Definition** | |
| This type of modelling capability derives from pooling mechanisms which:<br>- do not require identical data schemas<br>- proceed by linking multiple datasets (i.e. in the case of relational databases, by performing a JOIN operation).<br>The linkage mechanism may use any cross-referencing method (e.g. in the case of the Tech Sprint PoC, exact or fuzzy match on both counterparties in a financial transaction).<br>The join operation itself may be either a standard SQL join, or a more advanced one such as the second query example given in the PoC requirements document (namely, capturing transactions involving a common account and not further apart than 2 months). | |
| **Benefits and limitations** | |
| Joined data modelling is far more powerful than stacked data modelling because it enables a much richer feature set (predictive attributes) on a much larger universe (relations between transactions rather than simple transactions). Specifically, for a feature X present in datasets from A and B, such a model may take as input both features $X_A$ and $X_B$ simultaneously for joined transactions. | |
| **Detection models supported** | |
| Joined data modelling enables creating and tuning PDDMs which:<br>- use the union of data schemas as input features<br>- can be trained on the result of a join operation across training datasets (which is called vertical federated learning in the case of FL). | |

| **4. Account graph modelling** | Supported by **2** solutions |
|---|---|
| **Definition** | |
| This type of modelling capability assumes that the entire network of transactions has been built.<br>The most natural structure typically represents each account as a node and aggregates transactions at the edge level. | |
| **Benefits and limitations** | |
| Account graph modelling makes it possible to build both unsupervised and supervised detection models having a holistic view of the network of transactions (while still being compatible with more traditional table-based models). | |
| **Detection models supported** | |
| Account graph modelling enables creating and tuning PDDMs which the unification of the graph from each FI as input data for both training and testing. | |

### 4.4 PoC treatment by the solutions

This section presents in a bit more detail how each task in the Tech Sprint PoC was tackled and (fully or partially) solved by the 12 solutions.

#### Stage A (data pooling)

***Task 1: synthetic data generation***

The PoC requirements document described an optional synthetic data generation task: the goal was to generate an additional input dataset representing a fictional bank C's transactions, with the requirement that this dataset should have additional connections with A and B's datasets. This new dataset could have been generated synthetically properly speaking (i.e. by emulating certain features of existing datasets, most likely A and B's transactions), or pseudo-randomly - with the constraint that connections with A and B's data should yield interesting patterns, such as new suspicious activity patterns, which were not produced from those two datasets alone.

Half of the solutions presented chose to implement this optional task and thus included a synthetic data generation capability. The features proposed relied on various types of implementation:

- Some chose the random generation route by using an ad-hoc, statistical modelling-based approach.
- Others applied DP to input data.
- Some implemented a more robust approach using GANs (Generative Adversarial Networks).
- Yet others used a combination of DP and GANs, namely by leveraging a PATE-GAN framework which provides tight DP guarantees for a model trained on a GAN[7].

***Tasks 2, 3, 4: data upload and data pooling***

The variety of technical approaches proposed by Tech Sprint participants encourages to group the data upload and data pooling tasks together as they are typically performed as an atomic sequence or even simultaneously.

It should also be noted that some solutions do not even contain a data pooling step properly speaking:

- SMPC-based solutions typically create a virtual data pool whereby datasets are not brought to the same location but only bits and pieces (in some cases, down to individual bytes) will be randomized and encrypted by a secret sharing protocol.
- FL approaches operate by sharing detection models and not the training data fed to those models.
- As detailed in the next section, a few solutions perform ad-hoc queries on individual data items. This lies out of the Tech Sprint PoC scope and does not require any preliminary data pooling, however all but one solution provide this capability as an extra feature (in addition to model creation and training on pooled data, which was a required PoC capability).

The requirements document gave participants all freedom to choose the filters to apply during pooling. However, since choosing inadequate – particularly excessively broad – filters might lead to a combinatorial explosion (Cartesian product of the input datasets) for certain types of structures such as materialized databases, suggested filters were provided in the form of a few basic rules that could trivially be expressed as SQL queries:

[7] Bibliographical reference : Jinsung Yoon and James Jordon and Mihaela van der Schaar. PATE-GAN: Generating Synthetic Data with Differential Privacy Guarantees, International Conference on Learning Representations (2019).

- Join all legs of a transaction, i.e. the view of a given transaction from FIs A and B (and C if present).
- Join all transactions involving a common account and not further apart than 2 months.

An insightful observation from the overall PoC work by participants is that sophisticated linking of data sources proved very difficult to guarantee in a confidential setting. The more simple the pooling mechanism, the more likely it was to have been adopted by participants. The kind of pooling mechanism supported by a solution also directly affects the type of PDDM that can be created and tuned using the same solution.

Conversely, for graph-based structures, a general guideline is to load as much of the input data sources as possible into the pooled data graph as run-time storage and querying requirements for such structures tend to vary proportionally with the sum (not the product) of the input dataset sizes. The only participants which adopted this approach chose indeed to construct and analyze the entire graph (either of transactions, or of accounts which already provides one level of aggregation and thus higher intrinsic scalability).

### Stage B (model creation and tuning)

In a second stage (tasks 5 to 7), Tech Sprint PoC solutions had to enable FI A to upload its current detection model, called for the purposes of this experimentation a Solo Data Detection Model (SDDM) into the Confidentiality Perimeter. The choice had indeed been made of a simpler, more consistent experimental protocol wherein inference and evaluation would run as the same task on both solo data and pooled data.

The solution then had to enable FI A to build a new detection model called a Pooled Data Detection Model (PDDM) operating on pooled data. Participants were free to choose to implement either a rule-based model or an ML-based model (or both); some also chose to implement a graph-based model. PDDM inference, followed by the evaluation of its predictive performance, would then be performed in stage C on the chosen model type(s).

In the case of an ML-based PDDM, solutions had to support training the model on a subset of the input data; in the case of a rule-based PDDM, they had to support defining a set of custom detection rules appropriate for the PoC.

***Model training and target variables***

The PoC test datasets also included target variables in the form of an is_alert_solo (respectively is_alert_pooled) label associated to each transaction, indicating whether a transaction seen by FI A had been hypothetically flagged as suspicious based on A's view of the data only (resp. based on A's data augmented with its counterparty's data). The goal in task 5 was thus to optimize the SDDM to fit is_alert_solo, while the goal in task 7 was to train the PDDM to fit is_alert_pooled.

Those two target variables were not randomly generated, but did not either represent "source of truth" labels stemming from a real-world scenario using actual transactional data. The point of the PoC was not to measure the predictive performance of a model, but to demonstrate that a detection model (for instance, a suspicious activity detection model) could be integrated and trained inside the Confidentiality Perimeter.[8]

[8] The reason behind this choice is that there are only two ways to build a privacy-preserving predictive modelling solution operating on real-world data and in a realistic setting: either starting with real data

***Task 5: SDDM integration, creation and tuning***

Tech Sprint participants were asked to support both:

- The creation of an SDDM, where the model could be a very simple ML model since optimizing model performance was outside the context of the PoC.
- The upload and integration of a pre-trained model into the Confidentiality Perimeter.

Those tasks were marked as mandatory for the PoC because SDDM model integration, creation and tuning are supposed to be easier to implement than their PDDM counterparts. Their main goal was to provide a consistent solution for running inference on both model types (SDDM and PDDM) and comparing their performance[9].

***Tasks 6 and 7: PDDM creation and tuning***

The PoC requirements document explicitly suggested to consider both rule-based models and ML models as PDDMs; in addition to these two categories, some Tech Sprint participants chose to implement graph-based inference models.

Detection model creation and tuning was outside the focus of the Tech Sprint, whose primary goal was to provide a confidential computing solution, i.e. a turnkey service enabling to perform an entire collaborative analytics workflow with confidentiality and integrity guarantees. In other words, regarding the detection model–related stages of such a pipeline, the requirement was for the platform to support creating and optimizing a model, not to provide ML or AutoML[10] capabilities. Nevertheless, many Tech Sprint participants included basic or advanced ML functionality in their proposal, either because their off-the-shelf solution already embedded such functionality, or because they deemed it provided significant added value to their Tech Sprint PoC solution.

The general guideline for tasks 6 and 7 was to have as few moving pieces as possible between the baseline model (SDDM) and the pooled data model (PDDM). This does not imply that the SDDM and PDDM will be identical - they will at a minimum differ because the PDDM takes a larger feature set as input. And of course, feature importance would be adjusted to take into account the additional information (in the case of an ML-based model), and/or the rules would themselves be adjusted (in the case of a rule-based model). On the other hand, the goal was (both for the PoC work and for any potential experimentation performed later on real data) to avoid a situation such as a simple decision tree as SDDM and a neural network as PDDM, which would likely skew the measurement of the performance improvement associated to data pooling.

---

and showing that a model can be trained with good predictive performance and confidentiality guarantees all at once; or alternatively, start by addressing the secured pooling question by building a CDP solution, and only then feed real data into the PDDM to improve its actual predictive performance compared to an SDDM. The ACPR Tech Sprint clearly adopted the latter approach by design, after observing that other initiatives in the AML/CFT and fraud detection space had failed in addressing the former.

[9] Note that in principle, not only model creation and tuning but also inference on solo data could happen outside the Confidentiality Perimeter. The main reason for running SDDM inference inside the Confidentiality Perimeter is to ensure that the comparison between SDDM and PDDM performance is an apples-to-apples one, as well as provide a homogeneous developer experience when using the CDP platform.

[10] Automated machine learning (AutoML) is the process of automating the tasks of applying machine learning to real-world problems.

The ML functionality represented by the solutions covers an extremely broad scope:

- In terms of model classes, the solutions comprised unsupervised ML models, supervised ML models, rule-based models, or less typically causal models[11].
- Some solutions only supported a small number of ML model classes: in particular, HE-based solutions are not directly amenable to complex models and typically only support variations of linear models. However, some HE-based solutions enable training and inference not just of regressions, but also GLMs (Generalized Linear Models), and even GBTs (Gradient-Boosted Trees) or multi-layer perceptrons.
- In terms of model structure, some solutions operated on tabular data (mostly supervised ML models in this case), others on graph data (mostly unsupervised ML models in that case), and a few solutions had a sufficiently flexible API to support both kinds of models, source items and feature sets.

As explained in Section 4.2, each solution's confidential model creation and tuning capability directly derived from the type of data pooling mechanism implemented in the solution.

### Stage C (predictive performance)

***Tasks 8, 9, and 10: evaluation of models' predictive performance***

The PoC requirements document asked from Tech Sprint participants to compute performance metrics both for the baseline model operating on solo data (the SDDM) and for the new model operating on pooled data (the PDDM). In addition to standard metrics (accuracy and recall), participants were invited to provide any additional relevant metrics.

One of the most atypical characteristics of the CDP Tech Sprint (and one of the most counter-intuitive, even to participants) was that predictive performance did in no way constitute a success metric. In essence, the Tech Sprint aimed to select solutions most suitable to support confidential data pooling and model tuning, whereas model optimization should be a completely independent building block of the AML/CFT solution considered – and as such was out of scope for the PoC. Conversely, predictive performance values computed on fictitious data would not themselves be meaningful.

Some participants however opted to demonstrate their capability (and in some cases their tradecraft) in model optimization. They mostly compared metrics such as F1 score between SDDM and PDDM, which raised a number of caveats (they were free to choose model classes in both cases, the target variable on solo data did not necessarily map to target variable on pooled data, etc.) but in the majority of cases demonstrated a net gain in predictive performance resulting from data pooling.

A particularly interesting work was performed by two participants using unsupervised graph-based analytics, which led them to valuable insights in terms of potentially false positives (resp. false negatives) when comparing their results to the target label within the PoC datasets. This achievement was again out of the PoC scope, but showed that these two participants provide an "all-in-one" solution spanning data pooling, supervised machine learning and unsupervised analytics on confidential data.

[11] A causal model (or structural causal model) is a model that describes the causality mechanisms of a system. Causal models can improve study designs by providing clear rules for deciding which independent variables need to be included/controlled for. Causal models have, among other domains, found applications in Machine Learning. Within Tech Sprint solutions, those models were more precisely causal Bayesian networks, which combine expert knowledge with historical data, enable computing risk using uncertainty information, and have the added benefit of being convertible to rule-based models.

***Complementary tasks: output sample extraction and manual review***

Although manual review was strictly speaking not part of the PoC task assignment, the experimentation on real data following the CDP Tech Sprint will involve a manual review step, in which human annotators will label both detection model outputs. Some input variables will be visible during human review, whereas others (including some predictive variables used by the model) will not be available to reviewers because they carry sensitive information belonging to a counterpart institution. For the purposes of the PoC, a rudimentary, automated performance evaluation mechanism shall be used instead, but an output sample extraction task was required from proposed solutions.

Many solutions were lacking in this regard and had not implemented task 10.a, since no output export capability was made available. Others provided very basic export in which sensitive attributes were simply suppressed (which would only enable a minimalistic manual review, based on solo data and not on pooled data). A few however provided interesting export capabilities, either reducing the re-identification risk by applying basic pseudonymization or even quantifying the residual risk by using methods such as k-anonymity criteria and/or differential privacy.

# 5. Tech Sprint solutions in detail

This section presents each of the 12 solutions in more detail than the previous ones. The main characteristics are summarized along four dimensions: architecture of the solution, security model and its governance, the data pooling functionality itself, and the AML/CFT capability in terms of detection model creation and tuning.

For each solution, a link to a YouTube video presenting the company or companies and the solution in a couple of minutes is also given. English subtitles are provided for each video.

Those 12 video presentations, along with a 5-minute documentary summarizing the entire Tech Sprint, are grouped within a dedicated YouTube playlist:

➔ **Link to the Tech Sprint documentary on YouTube**

## 5.1 Atos - Privitar - TigerGraph – IBM

➔Link to the video presentation

| Solution architecture | |
|---|---|
| **Confidentiality perimeter** | **Hosting** |
| The confidentiality perimeter comprises an isolated Virtual Private Cloud (VPC) environment with IBM Secure FS Cloud hosting other nodes (including the Privitar platform and TigerGraph analytics). | IBM FS Cloud is the most secured cloud but it can also be run on any other cloud |
| **Participants and their roles** | |
| - An intermediary: a trusted third-party which acts as a gatekeeper for data processing (and does not have access to raw data)<br>- Atos: acts as the service provider for the CDP platform (and does not have access to raw data either)<br>- Data providers: each FI providing its own dataset | |

| Security model and governance |
|---|
| **Threat model** |
| The solution includes coordinating contributors with an intermediary who acts as a trusted third party. All contributors must therefore agree on the designation of the legal entity acting as the intermediary. |
| **Key management and encryption process** |
| When data is sent to the protected area, it undergoes double encryption: the first with a key of the contributor, the second with that of the intermediary.<br>Subsequently upon receipt:<br>- Data is decrypted using only the private key of the intermediary. At this stage, data is still protected by the contributor's encryption key.<br>- Data is re-encrypted with a so-called “blinding” key used for homomorphic encryption which allows the integrity of the data to be rebuild with consistency across the transmissions from all contributors.<br>- Data is then processed variable by variable according to the requirements of the FIs and the many PET techniques offered by Privitar (RegEx Tokenization, Disruption, Generalization, etc.). |

| Data pooling functionality | |
|---|---|
| **Data upload mechanism** | **Data pooling mechanism** |
| Each participating FI's (contributor) dataset is doubly-encrypted by Privitar software and uploaded to a VPC in the IBM Secure FS cloud. | Row matching (based on PSI) |
| **Fuzzy matching** | |
| Entity resolution and recognition based on graph analysis | |
| **Synthetic data generation** | |
| Both basic and ad-hoc (i.e. customized for a specific data schema) | |

| Detection modelling (and querying) functionality on pooled data | |
|---|---|
| **Input features supported** | **Confidential modelling capability** |
| Either the table of transactions, or the graph of transactions enriched with insights from graph analytics | Graph-based or not (PyTigerGraph, PyTorch Geometric, Tensorflow, DGL, etc.). Thus the models can be rule-based models on tabular or on graph data, Machine Learning models on tabular data, graph-based Machine Learning models, or combinations thereof. |
| **API** | |
| A Python data science environment with access to scikit-learn, PyTorch and PyTorch Geometric (for models operating on graph data), Tensorflow and DeepGraph Library. | |

## 5.2 Cleyrop - Cosmian – Ekimetrics

➔Link to the video presentation

| Solution architecture | |
|---|---|
| **Confidentiality perimeter** | **Hosting** |
| The confidentiality perimeter comprises a number of Data Hubs (deployed at each participating FI), a Pooling Hub including a TEE into which various datasets are centralized, and a Compute Hub on which model training and inference are performed. | OVH |
| **Participants and their roles** | |
| Cleyrop / Cosmian's TEE-based solution defines the following roles:<br>- TEE deployment environment: a trusted third-party for the experimentation<br>- Computation owner: administrator role for the experimentation<br>- Data providers: each FI providing its own dataset<br>- Code provider: the FI running its protocol on its own detection model<br>- Result consumer: the same FI, which is the only one allowed to view results. | |

| Security model and governance |
|---|
| **Key management and encryption process** |
| Most cryptographic primitives used in the solution are in the NaCl family:<br>- X25519 (RFC 7748) for the Elliptic Curve Diffie-Hellman (ECDH) key exchange<br>- Ed25519 (RFC 8032) for digital signatures<br>- XSalsa20-Poly1305 (RFC 7539) for the “Authenticated Encryption with Associated Data” (AEAD). |
| **Threat model** |
| The Secure Computation solution provides each participant with the following guarantees:<br>- They perform computations on a secure hardware enclave (a quote is generated from the enclave using a remote attestation process and guaranteeing that it is actually signed by a valid SGX enclave and not some simulator)<br>- The enclave operator (Cosmian) has not tampered with any computation parameters<br>- Both code and data can be read only by the secure enclave and neither by the operator, the hosting service provider, nor any other participant<br>- The list of participants is correct<br>- The code has the expected digital signature<br>- The rest of the software stack is correctly identified and thus verifiable. |
| **Query authorization** |
| The code provider must validate the computation. In addition, the secure enclave identity enables every participant to verify that the computation is executed inside a proper secure enclave. |

| Data pooling functionality | |
|---|---|
| **Data upload mechanism** | **Data pooling mechanism** |
| All participants start by sharing a secret. Once each participant has sent their public key to the secure enclave, the enclaves generates its ID, then the participant sends it encrypted data, and also sends its symmetric key sealed with the enclave public key and signed with its private key. | Tuple unification |
| **Fuzzy matching** | |
| Fuzzy matching (based on a Levenshtein distance) is executed within the secure enclave. | |

## 5.3 Decentriq

➔ Link to the video presentation

| Solution architecture | |
|---|---|
| **Confidentiality perimeter** | **Hosting** |
| The confidentiality perimeter is materialized as a Data Clean Room created by Decentriq's solution. | By default on EU-based Azure Confidential Compute instances. Also possible on a private cloud with TEE or on premise. |
| **Participants and their roles** | |
| - Platform making the Data Clean Rooms available, comprising TEEs and managed by Decentriq<br>- Data providers: each FI providing its own datasets<br>- Data analysts: users from FIs who analyze the data pool and retrieve results from the Data Clean Rooms | |

| Security model and governance | |
|---|---|
| **Technical details (algorithm or implementation)** | |
| The confidential computing technology is hardware-based and does not require ad-hoc algorithms built in the software layer to compute or process the data. Supported Confidential Computing resources are Intel SGX and AMD SEV-SNP. | |
| **Threat model** | |
| Assumes that a remote attacker has gained full remote access to the Decentriq platform, including the infrastructure and other cohabitants of the Confidentiality Perimeter and enclave itself. This threat model leverages the security properties of the TEE technology and its capability to attest remotely what is run to bring the root of trust down at the hardware level. It should be however noted that if an attacker gets physical access to the hardware, there is the chance that a physical manipulation of the CPU can leak any of the cryptographic key used. Therefore well-established and certified cloud providers should be used to protect the machines used by the solution. Thus the root of trust ultimately lies with the hardware manufacturers (Intel or AMD).<br>Furthermore, the logic that is run within the Confidentiality Perimeter is defined by the participants, and all input data is provided by participants. Thus a faulty logic or input data manipulation from one of the participants may lead to the possibility of leaking sensitive data. Therefore all participants should carefully review the Data Clean Room definition before uploading data and participating in the experimentation, and implement controls to ensure that the data they upload is correct. | |
| **Query authorization** | **Key management and encryption process** |
| The only processing that can happen on the data is what has been defined in the Data Clean Room and approved during the data upload by the respective data owners. | The key-exchange protocol relies on ephemeral keys. The encryption key used by the user to encrypt the data is generated during data upload, may be unique to each provisioned dataset, and is no longer needed once data is encrypted and provisioned to a Data Clean Room.<br>The key storage scheme works as follows:<br>- The user first encrypts the data encryption key with an encryption key provisioned by the enclave via a secure channel<br>- This encrypted key is then sent to the enclave<br>- The enclave receives the key, decrypts it and encrypts it again with another key that is known only by this specific enclave<br>- Finally, the enclave stores the encrypted key in a permanent storage that is deployed in the same data center as the enclave (e.g. Azure CH, Green.ch). |

| Auditability and traceability | Legal or expert opinion, certification |
|---|---|
| An audit log is generated within the confidential computing environment and allows full auditability and transparency on the actions performed in each Data Clean Room.<br>Whenever computations execute, errors at runtime get reported directly to the user while also preserving data confidentiality. | - customer data collaborations between FIs and news publishers for cookieless marketing in EU - legal opinion from MLL law firm confirming adherence to GDPR<br>- patient data collaborations between hospitals and pharma - legal opinion confirming adherence to GDPR<br>- secure collaborations on cyber threat intelligence amongst multiple FIs orchestrated by ArmaSuisse<br>- trade finance collaborations between shipping companies and corporate banks |

| Data pooling functionality | |
|---|---|
| **Data upload mechanism** | **Data pooling mechanism** |
| Each input dataset can be uploaded by the authorized user from the FI using either a Web UI or a Python SDK.<br>The data encryption happens locally and the encryption key is not shared to any other party, not even Decentriq.<br>The transmission of the encrypted copy of the data is secured via TLS and happens only after having recognized the server as secure and trustworthy via the TEE remote attestation protocol. | Any SQL query on an external SQL engine, or any Python script. The solution comprises a proprietary SQL engine which is ANSI SQL based on SQLite syntax and supports a limited subset of functions to provide strong guarantees on the computations and on the output privacy. |
| **Fuzzy matching** | |
| Python confidential VMs enable users to design or choose arbitrary fuzzy matching logic.<br>In Decentriq's proprietary SQL engine, a built-in function also allows to perform fuzzy matching. | |
| **Synthetic data generation** | |
| Yes, using PATE-GAN | |
| **Export functionality** | |
| Yes, using k-anonymity filter and DP | |

| Detection modelling (and querying) functionality on pooled data | |
|---|---|
| **Input features supported** | **Confidential modelling capability** |
| Table of transactions, enriched with account-level statistics | Rule-based or ML-based, using any Python code |
| **API** | |
| Users can interact with the Decentriq platform using either a web interface or Python and JavaScript SDKs. | |

## 5.4 Inpher

➔ **Link to the video presentation**

| Solution architecture | |
|---|---|
| **Confidentiality perimeter** | **Hosting** |
| The virtual confidentiality perimeter comprises an Inpher XOR machine installed at each participating node along with an orchestrator XOR service. | Cloud or on premise, both for the orchestrator and the nodes |
| **Participants and their roles** | |
| - XOR orchestrator service: the setup does not require a trusted third-party as the router never sees data and is self-managed<br>- Data providers: each FI providing its own dataset | |

| Security model and governance |
|---|
| **Technical details (algorithm or implementation)** |
| Inpher's core solution is based on SMPC and as a platform enables the use of other PETs such as Federated Learning (FL), Differential Privacy (DP) and Homomorphic Encryption (HE). The flexible, federated architecture allows for Virtual Machines (or AMIs) called XOR machines to be deployed virtually anywhere without specialized hardware.<br>The SMPC protocol used in the Inpher XOR Secret Computing platform is based on secret sharing and Fourier approximation of real-valued functions. For more details, see Inpher's peer-reviewed paper from Financial Cryptography 2018 and the Manticore paper. |
| **Threat model** |
| Honest-but-curious with a trusted dealer, in a full threshold setting. |
| **Legal or expert opinion, certification** |
| Independent EDPB Recommendations and Legal Analysis support the use of SMPC for compliant data transfer and collaboration under GDPR. |

| Data pooling functionality | |
|---|---|
| **Data upload mechanism** | **Data pooling mechanism** |
| Data can remain in its original source location and privacy zone, it is possible however (depending on the security requirements, e.g. if an institution would like not having to trust the cloud service provider) to encrypt the data and upload it into a secure enclave where the data resides together with the XOR machine. | Row matching, based on distributed SMPC processing (PSI using OPPRF + fuzzy matching using MinHash) |
| **Fuzzy matching** | |
| Yes, based on MinHash | |
| **Synthetic data generation** | |
| Yes, using DP | |

| Detection modelling (and querying) functionality on pooled data |
| --- |
| **Confidential modelling capability** |
| Linear/logistic Regression, XGBoost, k-Means |
| **API** |
| - XOR-py: a Python client library used to interact with the XOR Backend. It is a wrapper around the REST API of the XOR Platform which leverages higher-level abstraction for ease of usage. One can also choose to use the library for its lower-level primitives to interact with the full range of the XOR platform API.<br>- XOR REST API: for integration and interaction with other programs and applications via the RESTful web services.<br>- XOR DSL: a privacy-preserving composition library supporting local computations on private datasets, SMPC as well as FL computations with secure aggregation.<br>- XOR GUI: a graphical user interface providing similar functionality to the APIs. |

## 5.5 Roseman Labs

➔Link to the video presentation

| Solution architecture | |
|---|---|
| **Confidentiality perimeter** | **Hosting** |
| The confidentiality perimeter is virtual: it comprises the Virtual Data Lakes (VDL) located at each participating node and operating via secret sharing. The underlying technology is secure Multi-Party Computation | Any cloud |
| **Participants and their roles** | |
| - At least 3 VDL (ie MPC) nodes: the server nodes participating in the computation. To each such node, many clients (like a data owner or analyst) can connect.<br>- Data providers: each FI providing its own dataset<br>- A representative of the participants is a natural person that is given the authority to authorize queries | |

| Security model and governance |
|---|
| **Threat model** |
| The confidentiality of the data is protected via secret sharing during all computations. The VDL provides this confidentiality guarantee in the presence of malicious (actively corrupted) clients and honest-but-curious MPC nodes (passively secure MPC). |
| **Query authorization** |
| The query-authorization layer ensures that a client can only perform queries that have been explicitly approved by one or more representatives of the participants. The authorization mechanism is flexible in that it can authorize specific queries or entire query classes via a templating tool. The query-authorization layer serves as a powerful security control. For example, it will prevent uncontrolled data leakage in a "rogue employee" scenario, in which an employee of one organization tries to steal or browse through the entire data set from another organization. |

| Data pooling functionality | |
|---|---|
| **Data upload mechanism** | **Data pooling mechanism** |
| Upload via cryptographic mechanism such that the data arrive at the MPC nodes in secret-shared form only. | SQL outer join (demonstrated) and other types of joins (inner join, left join, fuzzy join etc.) |
| **Export functionality** | |
| Any, as long as approved by the participants that a certain type of calculation may be performed and to whom it can be disclosed | |

| Detection modelling (and querying) functionality on pooled data | |
|---|---|
| **Input features supported** | **Confidential modelling capability** |
| Table of transactions, or any other tabular data (including relational data bases) | Basic arithmetic and SQL-like operations, and more advanced statistical analyses like binary logistic regression. In the near future, multinomial/ordinal logistic regression, secure ridge regression, neural nets, and decision trees will also be available. |
| **API** | |
| A Python interface enables data analysts to interact with the VDL as if it were a normal database. Also an API library is available for direct interaction with the VDL | |

### 5.6 Sarus - Microsoft – EY

➔ Link to the video presentation

| Solution architecture | |
|---|---|
| **Confidentiality perimeter** | **Hosting** |
| The confidentiality perimeter is materialized and comprises a Microsoft Azure tenant, which hosts an Azure Key Vault Managed HSM for key management. | Microsoft Azure |
| **Participants and their roles** | |
| - An intermediary: the solution (including the key management and Sarus components) deployed on Azure but without any access to raw data<br>- Data providers: each FI providing its own dataset<br>- Data analysts or scientists able to access pooled data or alerts with the re-identification risk reduced by DP | |

| Security model and governance | |
|---|---|
| **Technical details (algorithm or implementation)** | |
| Data confidentiality relies on the following elements:<br>- Confidentiality and access control for cipher keys<br>- Confidentiality of process execution using a secure compute environment<br>- The use of remote attestations to authorize processes running in the enclave to access cipher keys. | |
| **Threat model** | |
| - Input privacy is addressed by using a secure enclave, along with key management by the FI<br>- Output privacy, i.e. the security of results from collaborative computations, is addressed by the Sarus solution. The threat model pertains to requests done through the API by (i) data scientists (ii) developers (iii) IT staff. Human operators or software processes may in those 3 situations send requests which potentially extract information relative to a peer FI's clients. The Sarus solution enables to minimize the associated re-identification risk. Observables considered are data attribute legitimately accessible via the API, whereas timing attacks (which leverage a correlation between execution times and the data) are ignored in this threat model. | |
| **Query authorization** | **Key management and encryption process** |
| Any outgoing information needs to be compliant with the data owner's security policy. If the strictest policy is adopted (DP with a fixed budget for a user group), those guarantees become the strongest. However an excessively strict policy may yield too much noise in output data.<br>The user may also authorize exceptions to the query mechanism (e.g. to access model performance measures, model weights, or inference values on a given client) for the benefit of specific users. | Data at rest is encrypted according to the storage mechanism adopted:<br>- Original data remains encrypted using the data provider's key and is stored in the secure enclave subjected to participants' security policy<br>- A Microsoft Azure storage account with the Storage Service Encryption (SSE) activated by default is used, whereby data is systematically encrypted using AES 256, making the process compliant with the FIPS 140-2 standard. Those AES keys are wrapped using an asymmetric key stored in the Microsoft Azure Key Vault Managed HSM. Participants can rely on their own key rather than a default key generated by Microsoft – a setup known as BYOK (Bring Your Own Key). |

| Data pooling functionality | |
|---|---|
| **Data upload mechanism** | **Data pooling mechanism** |
| Each participating FI's dataset is locally encrypted and uploaded to a Confidential Computing instance on the Microsoft Azure cloud. | Row matching, but in the clear |
| **Fuzzy matching** | |
| Yes, on all datasets behind the API. Confidentiality guarantees are still preserved, even after matching, in all information coming from the API. | |
| **Export functionality** | |
| Yes for every output that is compatible with the privacy policy (DP by default), including synthetic data samples. | |

| Detection modelling (and querying) functionality on pooled data | |
|---|---|
| **Input features supported** | **Confidential modelling capability** |
| Table of transactions including both legs (sender and recipient FIs) | Support for standard ML libraries:<br>- numpy and pandas are almost fully supported<br>- scikit-learn, TensorFlow and XGBoost are partially supported |
| **API** | |
| The Sarus API enables to interact with data in two days:<br>- By submitting SQL queries in the same dialect as the underlying SQL database (for the PoC this was Microsoft Azure T-SQL)<br>- By writing standard Python code using the Sarus SDK. | |

## 5.7 Scalnyx

➔ Link to the video presentation

| Solution architecture | |
|---|---|
| **Confidentiality perimeter** | **Hosting** |
| The confidentiality perimeter is the SCALTRUST platform onto which encrypted input datasets are uploaded. | Public (OVH, soon EXAION) or private cloud |
| **Participants and their roles** | |
| - Central service provider: the central server should be deployed and maintained by this neutral third-party for the experimentation. The central server runs autonomously and provides the services that must run in the Confidentiality Perimeter (i.e. in the encrypted domain where data are always encrypted, no FHE decryption key is accessible, and no decryption capabilities. The two services are "FHE querying as-a-service" (SQL-compliant) and "FHE and FL (and DP) as-a-service".<br>- Data providers: each FI which enrolls its dataset in SCALTRUST, uses one of the two services to perform computations and may also periodically update its dataset. | |

| Security model and governance |
|---|
| **Technical details (algorithm or implementation)** |
| The FL aggregator used in SCALTRUST makes computations exclusively with model parameters. These parameters are variable means, standard deviations for the built-in Causal-Naive Bayes model, and would be weights and gradients if we chose to use Neural-Network based models. (By contrast, with a full FHE ML approach, we would have to compute all the parameters required for the model directly over encrypted data, which is computationally very expensive to do within the FHE domain).<br>FHE Cryptosystem libraries used by SCALTRUST are bit-level BFV, SEAL, and Additive. |
| **Threat model** |
| - In the case of an honest but curious adversary: FHE-based aggregation is resistant but model privacy issues remain at the output (not an actual problem since the model is not supposed to be shared with other parties)<br>- In the case of a malicious party: output privacy issues and FL integrity issues<br>- FHE requires as usual the absence of collusion among participating nodes<br>- As each FI is in possession of the FHE decryption key, they must not access the central server that processes the encrypted data in FHE domain, otherwise compromising the confidentiality protection. In other words, the FI must not be the central service provider. |
| **Key management and encryption process** |
| The proposed solution does not address encryption key commissioning: all FIs are assumed to share the same keys through a secure key distribution protocol, and the keys are supposed to be generated locally by an elected FI. |

| Data pooling functionality | |
|---|---|
| **Data upload mechanism** | **Data pooling mechanism** |
| Each FI's sensitive data is directly sent then processed to the central server without ever interacting with FIs or other servers. | Model pooling through FL (FHE is used in model aggregation to protect parameters) |
| **Synthetic data generation** | |
| Via causal AI model | |
| **Export functionality** | |
| Also supported via causal AI model | |

| Detection modelling (and querying) functionality on pooled data | |
|---|---|
| **Input features supported** | **Confidential modelling capability** |
| Either the features common to all participants (horizontal FL setup) or the union of feature sets of each participant (vertical FL setup) | Causal AI Bayesian Network models.<br>Note that several scientific publications showed the feasibility of using Causal AI for fraud detection, however to the best of our knowledge, the only commercial product with a similar approach (Hugin AML) does not satisfy the Confidential Data Pooling Tech Sprint requirements. |
| **API** | |
| The platform supports three types of packaging:<br>- A platform SDK tool facilitates the integration of the SCALTRUST platform in a development environment (e.g. using Python)<br>- A “no-code” development tool enables the design of confidential applications by composing the available computing components, without any need for skills in cryptography or programming<br>- FHE development tools aid an advanced developer to implement new computing component in the homomorphic domain.<br>It also includes a SQL-compliant FHE query builder. | |

## 5.8 Secretarium – FutureFlow

➔Link to the video presentation

| Solution architecture | |
|---|---|
| **Confidentiality perimeter** | **Hosting** |
| The Secretarium Confidentiality Layer (SCL) is a component that ensures the security of the data end-to-end (e.g. in transit, at rest and during processing). The SCL stores data in encrypted tamper-proof ledgers and can run both rule-based or ML-based models for analytics. | Public cloud (OVH, SwissCom) or on premise |
| **Participants and their roles** | |
| - SCL including a TEE<br>- Each FI provides its dataset in encrypted form to the SCL | |

| Security model and governance | |
|---|---|
| **Technical details (algorithm or implementation)** | |
| Secretarium leverages secure hardware that is remotely attestable. The Secretarium Connection Protocol requires a remote enclave signature that embeds both the enclave thumbprint (the code) and various whitelisting guarantees (running a secure hardware, reviewed by Secretarium). This guarantees to each FI that they are effectively communicating to a specific reviewed and approved remote service that can be trusted. | |
| **Threat model** | |
| The solution performs de-identification inside the SCL, which eliminates the risk of accidental or deliberate re-identification in cases where the platform shares one FI's data with other FIs.<br>The Confidentiality Layer is integrity-driven and provides code attestation to ensure the participating FIs that only the pre-approved code will be applied to the data they supply. | |
| **Query authorization** | **Key management and encryption process** |
| No, except post-suspicion manual review | A secret key is generated and sealed automatically in the secure enclave, and is not visible by anyone, even to the participating FIs. |
| **Legal or expert opinion, certification, existing deployments** | |
| - Demonstrated since 2018 as part of the DANIE initiative<br>- Participation in ICO's regulatory sandbox<br>- Analytics applied at the FCA AML TechSprint in 2019<br>- Analytics applied on real-life data at Deloitte TriBank PoC in 2019<br>- Passed the largest Financial Institutions (FIs) IT Security audits<br>- Passed the security audit of an independent IT Security company<br>- Already running reconciliations with reference entity data | |

| Data pooling functionality | |
|---|---|
| **Data upload mechanism** | **Data pooling mechanism** |
| Before upload, data goes through multiple validations, integrity checks, and transformations stages on premise to ensure the quality of the reconciliations. | Graph-based matching |
| **Fuzzy matching** | |
| The solution is able to spot data errors automatically and do fuzzy matching, overcoming a major obstacle of blind coordinated one-way hashing, where the central counterparty has no way to identify data integrity issues. To prevent incorrect automated fuzzy matching, the fuzzy-matched accounts are proposed to financial institutions in secure rooms for manual resolution. | |
| **Export functionality** | |
| Pseudonymized export, which implies the absence of PII but nonetheless entails a residual re-identification risk | |

<table>
<tr><th colspan="2">Detection modelling (and querying) functionality on pooled data</th></tr>
<tr><th>Input features supported</th><th>Confidential modelling capability</th></tr>
<tr><td>The input dataset comprises transactions across de-identified accounts. The dataset is transformed to create a collection of graphs of de-identified accounts (with aggregated features computed across transactions). Input features for ML models are formal network properties of the graph of each account.</td><td>FutureFlow is based on unsupervised analytical methods. Cross-bank account relationships are analyzed as graphs, however no graph database is used, rather all graph data is stored in a highly scalable non-relational database and all graphs are generated on the fly on an ad-hoc basis for analysis and ranking.<br>LightGBM is also supported for supervised learning based on aggregate alert statistics, which provides a post-analysis evaluation step of unsupervised learning results.</td></tr>
<tr><th colspan="2">API</th></tr>
<tr><td colspan="2">No programmable interface, however the solution enables visualization and browsing of the account graph, highlighting owned/counterparty accounts, with suspicion flags, and the possibility of launching an investigation on-the-fly.</td></tr>
<tr><th colspan="2">Manual review</th></tr>
<tr><td colspan="2">This is the only one among Tech Sprint solutions that provides a robust manual review capability inside the confidentiality layer, which can be used both for pre-analysis fuzzy matching and data reconciliation, and for post-analysis joint investigations of the discovered cross-bank suspicious networks.<br>Reconciliations and investigations are done in secure rooms provided by the SCL, which are anonymously initiated by an FI:<br>- For data reconciliations, upon consent of every FI invited to the secure room, the fuzzy-matched data is revealed and the FIs can vote on matching the fuzzy-matched accounts. In the case of a match, the SCL re-processes the impacted pooled transactions prior to initiating the analytics steps.<br>- Post-analysis, a case can be created for joint investigation on a set of suspicious accounts, if necessary sharing sensitive information at a post-suspicion level.</td></tr>
</table>

## 5.9 SnowPack – Sphinx

➔ Link to the video presentation

| Solution architecture | |
|---|---|
| **Confidentiality perimeter** | **Hosting** |
| The confidentiality perimeter comprises:<br>- the Sphinx client installed at each FI (which hosts pseudonymized transactions)<br>- a number (>=2) of blind Sphinx servers on an infrastructure spanning multiple IaaS<br>- the Snowpack invisibility network overlay (SNO), used to fragment HTTPS packets and route them through different network routes | Any: Linux or Windows, LXC container or Docker, or a VM. |
| **Participants and their roles** | |
| - A number (>=2) of intermediaries, each a blind Sphinx server which relays requests and responses to the SNO (and does performance monitoring, usage reporting as well as some operational tasks)<br>- Data providers: each FI providing its own pseudonymized dataset through its Sphinx client | |

| Security model and governance |
|---|
| **Technical details (algorithm or implementation)** |
| Snowpack technology builds on 5 years of R&D and is protected by 3 patents for which it has a quasi-worldwide exclusive licensing agreement. SNO is operational and deployed across 4 EU countries, 4 cloud operators and 30 servers by end 2022.<br>Sphinx technology builds on 3 years of innovation within the Cybersec4Europe EU project and leverages several privacy-enhancing technologies (SMPC, homomorphic encryption, differential privacy combined with synthetic data generation) |
| **Threat model** |
| Snowpack supports the strongest threat model of all Tech Sprint solutions:<br>- Forward perfect secrecy at node level (IP and path are concealed by Shamir secret-sharing from nodes)<br>- Its ring architecture provides security and integrity guarantees as well as proof of origin (data comes from a ring member)<br>- It is resistant to man-in-the-middle attacks thanks to distributed exit nodes (thus no trusted third-party at the network level)<br>- Source and destination IPs are concealed from server (thus no trusted third-party at the application level).<br>Sphinx preserves data ownership through decentralized storage and data-encrypted computations, and provides technical measures (data in-use encryption at the application level) and organizational measures (sharing protocol, key management) to comply with GDPR and bank secrecy. |
| **Key management and encryption process** |
| A Sphinx client pseudonymizes uploaded data with 3 different secrets:<br>- Hash salt: used to hash joining variables before encryption for pooling purposes (shared only with other network participants and not the Sphinx server)<br>- Private encryption key: used to encrypt joining variables before pooling data and after hashing it with the hash salt (specific to each network participant, stored in local, can be outsourced to a key manager)<br>- Shared encryption key: used to encrypt confidential insights before transit/transfer on network<br><br>Creation and rotation of shared secrets (hash salt, shared encryption key) are managed through a Diffie Hellman-based key exchange protocol and Shamir secret sharing to prevent any interception. The reset frequency is fully configurable from SPHINX console. |

| Data pooling functionality | |
|---|---|
| **Data upload mechanism** | **Data pooling mechanism** |
| Each participating FI's transactions are first pseudonymized before being loaded into the local Sphinx client. The decryption key is never stored. Once loaded, they are automatically available for confidential computations with other participant's data. | Row matching (based on PSI, batch matching (based on compressed and probabilistic data structures) |

| Detection modelling (and querying) functionality on pooled data |
|---|
| **Input features supported** |
| During data upload process, each client chooses which the sensitive queries to only use for join operations are, and which those to be used as predictive features are too. Depending on this choice, each piece information is encrypted to only be involved in simple computation like join operations, or to provide models with fresh data to train it or confirm its output predictions. |

## 5.10 ThetaRay – Duality

➔ Link to the video presentation

| Solution architecture | |
|---|---|
| **Confidentiality perimeter** | **Hosting** |
| Duality offers two different setups: a hub deployment or a peer-to-peer deployment. A hub offers several benefits (scaling up and down network participants more easily, aggregation and anonymization of results, value-added computations and network-level insights). Hence a hub deployment is recommended in all instances with more than 2 participants.<br>The confidentiality perimeter thus comprises a Duality node deployed at each participating FI, along with the optional central hub. | Linux OS (either Ubuntu 20.04 or RHEL 7.8) with Docker |
| **Participants and their roles** | |
| - Each FI participating as Data Owner (connect platform to view of data source, associate it with a project and define data approvals/permissions), Data Analyzer (train, tune, deploy and evaluate ML models or SQL-like queries), or even Model Owner (upload an ML model to the platform) runs a Duality node, which contains all the components required for successful execution of interwork participant tasks. Each participating FI can act in any number of these roles.<br>- Centralized Hub: a third-party operating as Orchestrator (establish and configure the collaboration project and the participants' roles, distribute the encrypted computation to relevant Data Owners, aggregate the response and sending it to the Data Analyzer), and optionally also execute the computation of data in use. The Hub is never exposed to any sensitive data, so any organization can act in this role, including a participant FI, a cloud provider, Duality, etc. | |

| Security model and governance | |
|---|---|
| **Technical details (algorithm or implementation)** | |
| The platform leverages the open-source and standards compliant OpenFHE fully homomorphic encryption library, and combines it with data science capabilities, which can be consumed via a UI or API.<br>Privacy-protected inquiries return results in seconds. Duality also participates in the HEBench.org initiative, a performance benchmarking framework that enables comparison across different implementations of homomorphic encryption. | |
| **Threat model** | |
| The type of encryption used - which is quantum-safe and probabilistic - ensures that every time the data is encrypted, the encrypted representation will appear different to a potential adversary, which prevents man-in-the-middle attacks (i.e. in this case, prevents a malicious party from corrupting the result that an AML/CFT analyst would receive). It is also known to be resistant to attacks from quantum computing devices. | |
| **Query authorization** | **Key management and encryption process** |
| Data owners are able to define and manage the specific permissions each party has in regards to queries. For example, they can define the types of queries, the types of responses (e.g., numbers, yes/no, retrieve, etc.), the number of queries, and more, that each participants and each specific role (e.g., analyst vs. manager) can execute. As such, by defining the appropriate query structure and responses, privacy and confidentiality can be preserved, and no unwanted information is leaked. | The platform exposes the Data and Key Management API, which provides methods of data and key storage and extraction, data lookup, as well as key life-cycle management. |
| **Auditability and traceability** | |
| Duality allows for encrypted computations in a scalable manner, storage and transfer of data, enforcement of data access restrictions, and auditing of the operations. | |

| Data pooling functionality |
| --- |
| **Data pooling mechanism** |
| Datasets can be encrypted at the Data Owner's premises and then connected as either joins or unions. As with the above, this collaboration mechanism also requires a mutually-agreed data schema. |
| **Fuzzy matching** |
| The Duality platform provides various data pre-processing capabilities, including fuzzy matching and normalization:<br>- Open pre-processing (users can upload any Python pre-processing script, with Data Owners having full control over what the script can perform, and being able to approve or deny any pre-processing of their data)<br>- Text normalization (to improve the matching rate using various proprietary and standard normalization and entity resolution algorithms)<br>- Any of these may be combined with any heuristic approaches. |
| **Synthetic data generation** |
| Possible using DP |

| Detection modelling (and querying) functionality on pooled data |
| --- |
| **Confidential modelling capability** |
| Duality supports a number of machine learning models, including GLM (Generalized Linear Models) and GBT (Gradient Boosted Trees).<br>One-off queries supported include arbitrary pandas queries, pre-approved SQL queries, and encrypted stat analysis or ML. |
| **API** |
| - Each Duality node is based on the homomorphic computation capabilities provided by the OpenFHE library, extended and wrapped by the object-oriented Computation API.<br>- The Computation API provides easy access to OpenFHE's homomorphic encryption capabilities. This API can be used by advanced developers for development of new data science methods for proprietary computation plugins.<br>- Another API exposed is the Data and Key Management API. This API provides methods of data and keys storage and extraction, lookup, data, and keys life-cycle management.<br>- The Data Science Layer exposes advanced Data Science capabilities: statistical, GBT and GLM models, etc. |

## 5.11 TripleBlind - Accenture

➔Link to the video presentation

| Solution architecture | |
|---|---|
| **Confidentiality perimeter** | **Hosting** |
| The confidentiality perimeter is called a Virtual Data Pool, and comprises the TripleBlind access point installed at each participating node. The centralized TriplerBlind router helps establish the Virtual Data Pool but is not part of the Virtual Data Pool. | Client hosted on their public cloud (GCP, AWS) or their on-premise environment. |
| **Participants and their roles** | |
| - TripleBlind router: the setup does not require a trusted third-party as the router never sees data and is self-managed<br>- Data providers: each FI providing its own dataset and their own algorithm | |

| Security model and governance |
|---|
| **Technical details (algorithm or implementation)** |
| TripleBlind's distributed computation keeps the data resident. When an approved analysis is performed:<br>- The data is one-way encrypted at the byte level (no encryption keys are employed, which could be misused or mismanaged)<br>- The analyzing algorithm or training model is also one-way encrypted<br>- Both parts are then split and pieces are distributed to all parties where encrypted computations are run<br>- The results are recombined and only the output is returned<br>- Neither the unencrypted algorithm nor the raw data ever leaves the owner's firewall. |
| **Threat model** |
| TripleBlind software is resident at the financial institution and therefore conforms to the cybersecurity and compliance posture defined by the institution. Security capabilities include the following:<br>- Upon approval by data owners, TripleBlind makes exploratory data analysis reports of pool participant's datasets available in order to help consumers understand the available data. These reports never show actual data, rather they show metadata about the dataset.<br>- Connections between sharing partners happen using secure ports over well accepted, encrypted transmissions.<br>- Access point software is hardened against third-party attacks, and this software resides behind an institution's firewall so within an already secure perimeter.<br>- Usage of data or algorithms by collaborative partners must be approved for every operation. Users can establish "agreements" or policies that provide for automatic approval. Data and algorithms are not left in third party platforms where unauthorized usage can occur. |
| **Key management and encryption process** |
| Method based on secret sharing, therefore no key management is necessary. |
| **Legal or expert opinion, certification** |
| - The EDPB (European Data Protection Board) considers that the split processing performed "provides an effective supplementary measure" to secure data (Recommendations, June 28th 2021, use case 5)<br>- Independent technical evaluation of SPMC technologies by MITRE (A US government funded research organization).<br>- TripleBlind has received a HIPAA Expert Determination stating that usage of TripleBlind meets and exceeds all US healthcare regulatory requirements (HIPAA) for de-identification of data at the time of usage.<br>- Legal opinions stating that TripleBlind provides GDPR-level anonymization of data |

| Data pooling functionality | |
|---|---|
| **Data upload mechanism** | **Data pooling mechanism** |
| Data is never uploaded to TripleBlind. Instead banks register the location of their data files. This is called "positioning" a dataset. In other words, data is positioned as an asset on a given access point, but always remains at rest in its original location. | Pooling only happens upon request. Does not require a mutually-agreed data schema. When pooling requests are approved by all parties (data pools can be created between 2 to N number of parties), TripleBlind's one-way encryption processes create the virtual data pool. |
| **Synthetic data generation** | |
| TripleBlind can be used to create synthetic data. The intent of the GAN synthetic generator in TripleBlind is to help the data consumer ( scientist/analyst) understand the format of the data to streamline their model building and analysis. Users benefit from perfecting their model testing on synthetic data. Users then benefit by getting actual results using undiluted, protected data. | |

| Detection modeling (and querying) functionality on pooled data | |
|---|---|
| **Input features supported** | **Confidential modeling capability** |
| Both horizontal and vertical combinations of data are supported. For the PoC, we trained a Knowledge Graph AI model to represent the "legs" of the transactions, and then used an XGBoost algorithm to identify suspicious transactions. We did this across all of the available data in a private fashion.<br><br>TripleBlind also demonstrated "blind-join" capabilities which make SQL-like commands available to safely query "virtually pooled" data across the physically federated data sets. | TripleBlind supports SQL (SQLite) and several R/Python data science libraries and capabilities.<br><br>TripleBlind supports Python data science libraries such as scikit-learn, pytorch, keras, TensorFlow, as well as PMML and ONNX capabilities to work with models and analytics from a wide range of toolsets. |
| **API** | |
| TripleBlind has a REST-based API. It is programmable using various open-source Python ML libraries (scikit-learn, XGBoost, PyTorch, Keras, Tensorflow, Ampligraph…) | |

## 5.12 Tune Insight

➔ **Link to the video presentation**

| Solution architecture | |
|---|---|
| **Confidentiality perimeter** | **Hosting** |
| The confidentiality perimeter comprises the Tune Insight agent installed at each participating node and communicating with other participants (using "*Multiparty Homomorphic Encryption"* or MHE) and with the authentication service. | On-premise or on cloud, any VM with Docker |
| **Participants and their roles** | |
| - Data providers: each FI providing its own dataset. A Tune Insight agent is a server that runs at each data provider.<br>- A centralized authentication/authorization service connected to the local or federated identity/directory services for collaborations and thus ensuring that the network of collaborating organizations can define their own policies for authorizing various actions depending on their needs, and manage their own users and roles. | |

| Security model and governance | |
|---|---|
| **Technical details (algorithm or implementation)** | |
| Multiparty computations are implemented on top of the open-source FHE library Lattigo. When a computation is requested, the following sequence takes place:<br>1. The request is forwarded to and validated by all agents that are participating in the project. An audit log is produced for traceability purposes.<br>2. Each agent loads the data to be used as input from the appropriate data source and logs the query and metadata about the input, such as the number of data records that resulted from the query.<br>3. The agents engage in a MHE computation:<br>a. The required cryptographic keys are generated collectively. Typically, the agents generate a collective public key and each one holds a share of the associated secret key. These keys can be ephemeral or static (subject to periodic renewal as determined by the applicable FIs' information security policies).<br>b. The local datasets are encrypted under the collective public key and the encrypted result is jointly computed homomorphically.<br>c. The encrypted result is collectively re-encrypted to the public key of the authorized user(s) or the originating agent. This choice depends on the use case and whether or not the results should be decrypted on the client side or on the user's institution agent.<br>d. The result can be decrypted and used as input to further processing by the requesting user. | |
| **Query authorization** | **Key management and encryption process** |
| Yes, using Attribute-Based Access Control and automated/semi-automated access protocols embedded in the MHE computation. | The Tune Insight agent can be configured to connect to common Key Management Services such as Azure Key Vault to retrieve sensitive credentials for local access to database and storage. |
| **Auditability and traceability** | |
| By leveraging an immutable database with cryptographic verification using Merkle trees, Tune Insight agents' audit logs are stored locally with strong integrity guarantees, while avoiding complex blockchain solutions. Among other features, any creation or edition of logs is systematically saved and traced within the database, which enables the retrieval of detailed audit trails capturing any actions that performed throughout a process (e.g. pooled data retrieval, training protocol, or aggregation protocol). | |

| Data pooling functionality | |
|---|---|
| **Data upload mechanism** | **Data pooling mechanism** |
| The data upload step simply consists of connecting the locally-deployed Tune Insight agent to the FI's data source. To avoid storing credentials information on the agent, the solution also provides a means to connect to the organization's KMS to retrieve database credentials. | Multi-party distributed pooling (based on PSI). Data is virtually pooled, as no raw data is transferred.<br>Another feature is the computation of vertically-partitioned statistics on (virtually) pooled data. |

| Detection modelling (and querying) functionality on pooled data | |
|---|---|
| **Input features supported** | **Confidential modelling capability** |
| The solution supports both training on horizontally stacked datasets and training on vertically joined datasets. | The Tune Insight agent's MHE capabilities support the following analytics:<br>- Collective Encrypted ML for training regressions and Multilayer Perceptrons (MLP), and hybrid encrypted-differentially-private ML for secure federated training of complex models<br>- Matching protocols such as Private Set Intersection and Vertically Partitioned Statistics<br>- Encrypted Aggregate Statistics<br>- Encrypted ML Inference with support for many models (GLM, SVM, CNN), protecting both data and model<br>- Private Information Retrieval (PIR, protecting the performed query) |
| **API** | |
| Three modules provide interaction capability with Tune Insight's solution:<br>- Each Tune Insight agent is a service that exposes a RESTful HTTPS API to the local clients and communicates with other external collaborating agents.<br>- Tune Insight Web Interface: client web application that provides a graphical user interface to easily engage in collaborations and run secure data analyses.<br>- Tune Insight Python SDK: client library that provides high level functionalities that are meant to be easy to use for data scientists in order to integrate Tune Insight's functionalities into their usual workflows. | |

# 6. Glossary

***AML/CFT (Anti-Money Laundering/Combating the Financing of Terrorism)***: The financial sector is exposed to money laundering and terrorist financing (ML/TF) risks. As such, it is subject to anti-money laundering and anti-terrorist financing (AML/CFT) provisions. (Source: ACPR.)

***CDP (Confidential Data Pooling)***: this term (coined by the ACPR for the experimentation) aims to describe any technology that enables storing, joining, querying, and feeding to AML/CFT models a number of sensitive datasets. Sensitive datasets in this context mean any type of data - primarily transactional - with confidentiality, privacy and integrity requirements by all stakeholders in the experimentation. The scope of CDP was expected to have a significant overlap with PET (Privacy Enhancing Technologies, see below), however a dedicated term is used on purpose in order not to limit the range of techniques that can be proposed by applicants.

***Confidentiality Perimeter***: for the purposes of describing a CDP solution, this perimeter generically refers to a physical or logical data storage and analysis area that provides appropriate confidentiality guarantees. For example in the case of end-to-end encryption this would comprise the entire experimental protocol (both storage area and data flows) following initial encryption of sensitive data.

***DP (Differential Privacy)***: one possible family of techniques for performing CDP. Differential privacy is a system for publicly sharing information about a dataset by describing the patterns of groups within the dataset while withholding information about individuals in the dataset. The idea behind differential privacy is that if the effect of making an arbitrary single substitution in the database is small enough, the query result cannot be used to infer much about any single individual, and therefore provides privacy. (Source: Wikipedia entry.)

***FHE (Fully Homomorphic Encryption)***: a variant of Homomorphic Encryption (see entry on HE) which allows the evaluation of arbitrary circuits composed of multiple types of gates of unbounded depth and is the strongest notion of homomorphic encryption.

***FI (Financial Institution)***: also called participating FI within the context of the ACPR experimentation on AML/CFT, this term designates all entities with AML/CFT obligations that participate on a voluntary basis in the ACPR experimentation, including the key phase which is the CDP Tech Sprint. Those entities are mostly credit institutions, with a few insurers also involved. Those institutions are assembled in teams, each working on a specific use case. Within a team, each institution remains free to choose its own experimental protocol and the most appropriate technology provider, this latter point being precisely the purpose of the CDP Tech Sprint.

***FL (Federated Learning)***: one possible family of techniques for performing CDP. Federated learning (also known as collaborative learning) is a machine learning technique that trains an algorithm across multiple decentralized edge devices or servers holding local data samples, without exchanging them. This approach stands in contrast to traditional centralized machine learning techniques where all the local datasets are uploaded to one server, as well as to more classical decentralized approaches which often assume that local data samples are identically distributed. Federated learning enables multiple actors to build a common, robust machine learning model without sharing data, thus allowing to address critical issues such as data privacy, data security, data access rights and access to heterogeneous data. (Source: Wikipedia entry.)

***GAN (Generative Adversarial Network)***: a class of machine learning frameworks in which two neural networks contest with each other in the form of a zero-sum game, where one agent's gain is another agent's loss. Given a training set, this technique learns to generate new data with the same statistics as the training set. Though originally proposed as a form of generative model for unsupervised learning, GANs have also proved useful for semi-supervised learning, fully supervised learning, and reinforcement learning. (Source: Wikipedia entry.)

***GBT (Gradient-Boosted Tree)***: Gradient boosting is a machine learning technique used in regression and classification tasks, among others. It gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees. When a decision tree is the weak learner, the resulting algorithm is

called gradient-boosted trees; it usually outperforms random forest. A gradient-boosted trees model is built in a stage-wise fashion as in other boosting methods, but it generalizes the other methods by allowing optimization of an arbitrary differentiable loss function.

***GLM (Generalized Linear Model)***: in statistics, a generalized linear model is a flexible generalization of ordinary linear regression. The GLM generalizes linear regression by allowing the linear model to be related to the response variable via a link function and by allowing the magnitude of the variance of each measurement to be a function of its predicted value. (Source: Wikipedia entry.)

***HE (Homomorphic Encryption)***: one possible family of techniques for performing CDP. Homomorphic encryption is a form of encryption that permits users to perform computations on its encrypted data without first decrypting it. These resulting computations are left in an encrypted form which, when decrypted, result in an identical output to that produced had the operations been performed on the unencrypted data. Homomorphic encryption can be used for privacy-preserving outsourced storage and computation. This allows data to be encrypted and out-sourced to commercial cloud environments for processing, all while encrypted. For sensitive data, such as health care information, homomorphic encryption can be used to enable new services by removing privacy barriers inhibiting data sharing or increase security to existing services. (Source: Wikipedia entry.)

**Honest but curious (adversary)**: also called semi-honest, the honest-but-curious adversary is a legitimate participant in a communication protocol who will not deviate from the defined protocol but will attempt to learn all possible information from legitimately received messages. (Source: *Modelling and Automatically Analysing Privacy Properties for Honest-but-Curious Adversaries*, Andrew Paverd et al.)

***Input privacy***: input privacy and verification are guarantees about the inputs of an information flow. See output privacy for a related, complementary concept.

***KMS (Key Management System)***: system used for managing cryptographic keys in a cryptosystem. This includes dealing with the generation, exchange, storage, use, crypto-shredding (destruction) and replacement of keys. It includes cryptographic protocol design, key servers, user procedures, and other relevant protocols. (Source: Wikipedia entry.)

***ML (Machine Learning)***: any AML/CFT detection model relying on any class of ML (linear or logistic regression, random forest or gradient boosted trees, etc. – although probably not deep neural networks) or any combination of such classes.

***Output privacy***: output privacy and verification are guarantees about the outputs of an information flow. See input privacy for a related, complementary concept.

***Participant (Tech Sprint Participant)***: any technology provider who responded to the Call for Applications published on May 16, 2022 and was subsequently selected by the ACPR to participate in the Tech Sprint.

***PDDM (Pooled Data Detection Model)***: for an FI participating in the experimentation (or for the fictitious FI A in the Tech Sprint PoC), this means an AML/CFT detection model operating on its data augmented with data from one or several other FIs.

***PET (Privacy-Enhancing Technologies)***: also called privacy-preserving technologies, a family of techniques that embody fundamental data protection principles by minimizing personal data use, maximizing data security, and empowering individuals. PETs allow online users to protect the privacy of their personally identifiable information (PII), which is often provided to and handled by services or applications. PETs use techniques to minimize an information system's possession of personal data without losing functionality. (Source: Wikipedia entry.)

***Pooling Mechanism***: the method, and its implementation, used in the ACPR experimentation by a team of co-participating FIs to pool their respective datasets. In order to assess the entire improvement margin of data

pooling, the ACPR encouraged participants in the experimentation to explore the more sophisticated pooling mechanisms. Those might for example be based on any number of complex SQL queries including joins, and potentially fuzzy joins, as opposed to more traditional pooling techniques (e.g. simply multiple datasets with identical schemas, or sharing minimal information such as Boolean values representing a “suspicion” flag).

***PSI (Private set intersection)***: an SMPC (see next entry) cryptographic technique that allows two parties holding sets to compare encrypted versions of these sets in order to compute the intersection. In this scenario, neither party reveals anything to the counterparty except for the elements in the intersection. (Source: Wikipedia entry.)

***SDDM (Solo Data Detection Model)***: for an FI participating in the experimentation (or for fictitious FI A in the Tech Sprint PoC), this means an AML/CFT detection model operating on its own data.

***SMPC (Secure Multi-Party Computation)***: one possible family of techniques for performing CDP. Secure multi-party computation (also known as secure computation, multi-party computation (MPC) or privacy-preserving computation) is a subfield of cryptography with the goal of creating methods for parties to jointly compute a function over their inputs while keeping those inputs private. Unlike traditional cryptographic tasks, where cryptography assures security and integrity of communication or storage and the adversary is outside the system of participants (an eavesdropper on the sender and receiver), the cryptography in this model protects participants' privacy from each other. (Source: Wikipedia entry.)

***Trusted Execution Environment (TEE)***: one possible family of techniques for performing CDP. A trusted execution environment (TEE) is a secure area of a main processor. It guarantees code and data loaded inside to be protected with respect to confidentiality and integrity, Data integrity — prevents unauthorized entities from altering data when any entity outside the TEE processes data, Code integrity — the code in the TEE cannot be replaced or modified by unauthorized entities, which may also be the computer owner itself as in certain DRM schemes described in SGX. This is done by implementing unique, immutable, and confidential architectural security such as Intel® Software Guard Extensions (SGX) which offers hardware-based memory encryption that isolates specific application code and data in memory. SGX allows user-level code to allocate private regions of memory, called enclaves, which are designed to be pro.tected from processes running at higher privilege levels. A TEE as an isolated execution environment provides security features such as isolated execution, integrity of applications executing with the TEE, along with confidentiality of their assets. In general terms, the TEE offers an execution space that provides a higher level of security for trusted applications running on the device than a rich operating system (OS) and more functionality than a 'secure element' (SE). (Source: Wikipedia entry.)

# 7. Appendix: evaluation form template

Appreciation criteria were grouped in three sections according to the corresponding phase within the overall AML/CFT experimentation, which each section containing multiple questions, for a total of 10 questions (and each question in turn made up of 3 to 5 items).

## First section

This section solely pertains to works done for the PoC on fictitious data, performed within 3 months and presented on September 13.

***Question 1 – Security guarantees***

- Threat model
- Confidentiality guarantees for input data according to their sensitivity level
- Integrity guarantees for both code and data
- Estimation of the residual risk, security certifications (norms, standards, external audits or legal analysis)

***Question 2 – Technology***

- Type of technology (PET or other, software / hardware / hybrid)
- Scalability in phase A (data pooling)
- Scalability in phase B (detection modelling)

***Question 3 – Functional coverage of the solution***

- Data pooling functionality (exact or fuzzy matching, etc.)
- Analytical functionality (ML-based models, SQL rules, etc.)
- Evaluation of the predictive performance
- Traceability and auditability

## Second section

This section pertained to the solution (built of course on top of Tech Sprint works) considered for conducting the experimentation on real data planned for a 6-to-9-month period following the Tech Sprint.

***Question 4 – AML/CFT modelling***

- AML/CFT expertise (to complete an FI's in-house expertise or to test new approaches)
- Scalability in phase A (data pooling)
- Scalability in phase B (detection modelling)

***Question 5 – Functional adequacy with experimental protocols***

- Input data (variable types, schema, volumes)
- Data pooling mechanisms (operating on natural or moral persons)
- Detection model classes (supervised or not, relational or graph analysis, …)
- Model output (transaction triaging or routing, client risk score, weak signals, …)

***Question 6 – Technological adequacy***

- Appetite of each participating FI for innovation (i.e. adopting relatively immature technologies)
- Systems integration constraints (of internal or external applications) of each FI
- Deployment constraints imposed by each FI (compatibility with in-house or outsourced infrastructure, or with market standards)

***Question 7 – Data management adequacy***

- Data governance policy of each participating FI
- Hosting policy of each FI
- Legal framework interpretation by each FI (with respect to personal data protection, banking secrecy, …)

## Third section

This section proposed a cost/benefit analysis for the potential adoption of a given Tech Sprint solution. In other words the goal was to project the evaluation into a production (or even still experimental) application scenario going beyond the one-shot AML/CFT experimentation proposed by the ACPR.

***Question 8 – Added value for domain experts***

- Efficacy gains (reduction of false positive rate, detection of new ML/TF signals or patterns, …)
- Efficiency gains (acceleration of risk profile analysis, automation of the investigatory process, …)
- Evaluation of new AML/CFT detection models

***Question 9 – Operational impact***

- Impact on functional teams (functional maintenance of the solution, e.g. parameter settings)
- Impact on IT teams (bug fixes and feature changes in the application, …)
- Impact on Data and Innovation teams (intrusiveness of the solution, changes in developer workflow, limits imposed on supported AI models, model retraining capability, …)
- Impact on Security and Infrastructure teams (autonomy in the technical maintenance of the solution, process changes e.g. for encryption key lifecycle management, …)

***Question 10 – Cost and associated risks***

- Hardware and software cost
- Human resources (IT, RDBMS, Data Science, …)
- Solution maturity (market penetration, scope of each tech provider's involvement in case of a Tech Sprint partnership, …)
- Risks associated to the adoption (technical failures, legal hurdles, …)