Privacy and DRM Requirements for Collaborative Development of AI Applications

Vida Ahmadi Mehri
Blekinge Institute of Technology
vida.ahmadi.mehri@bth.se

Dragos Ilie
Blekinge Institute of Technology
dragos.ilie@bth.se

Kurt Tutschku
Blekinge Institute of Technology
kurt.tutschku@bth.se

ACM Reference Format:
Vida Ahmadi Mehri, Dragos Ilie, and Kurt Tutschku. 2018. Privacy and DRM Requirements for Collaborative Development of AI Applications. In *ARES 2018: International Conference on Availability, Reliability and Security, August 27–30, 2018, Hamburg, Germany*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3230833.3233268

Abstract: The use of data is essential for the capabilities of Data-driven Artificial intelligence (AI), Deep Learning and Big Data analysis techniques. This data usage, however, raises intrinsically the concerns on data privacy. In addition, supporting collaborative development of AI applications across organisations has become a major need in AI system design. Digital Rights Management (DRM) is required to protect intellectual property in such collaboration. As a consequence of DRM, privacy threats and privacy-enforcing mechanisms will interact with each other.

This paper describes the privacy and DRM requirements in collaborative AI system design using AI pipelines. It describes the relationships between DRM and privacy and outlines the threats against these non-functional features. Finally, the paper provides first security architecture to protect against the threats on DRM and privacy in collaborative AI design using AI pipelines.

1 INTRODUCTION

Collaborative application development across organisations has become a major focus in Data-driven Artificial Intelligence (AI) system design when aiming at sophisticated AI applications[1, 2]. This collaboration process builds on specialisation in AI engineering and on re-useable AI objects, e.g. data set or Deep Learning models. These objects have been gathered or developed by third-parties not designing the final application. The advantages of the process are potentially significant reductions of development cost and time and access to components that enable engineering for higher AI performance. The appealing features are evidenced by the development of AI pipelines [3], open source machine learning and data visualisation tools such as Orange [4] and the emerge of data marketplaces [5, 6].

This collaborative approach, however, comes at a cost. It imposes at least three fundamental challenges on the design process. First, the use of data intrinsically raises data privacy concerns. These doubts become even deeper regarding the feature of datasets being shared. Second, Data-driven AI aims at identifying unknown relationships within the information. However, when using typical privacy enforcing mechanisms such anonymisation techniques or restriction in data collection, then it can't be excluded that inherent relationships within the data sets are not captured or deleted. As a result, such data sets are becoming of no value. While such typical privacy concepts are of high value for specific applications, they might impact the usability of AI objects in general. Hence, a dilemma for the general concept of collaboration based on reusability arises. Third, the reuse of AI objects in collaborative design requires trust among the developers and users of these objects. This trust ranges from obeying licences between developers to permitting governance on AI objects as required by societies and individuals, e.g. enabling GDPR or GDPR-like concepts on the use of data and AI objects in Europe. This paper aims to address these fundamental challenges by giving the insight to the privacy requirements in collaborative AI development. It will provide an initial taxonomy of privacy and Digital Rights Management (DRM) and the threats against objects in the AI pipeline. The paper summarises the GDPR act and its potential implications for Bonseyes-like AI marketplaces and describes potential ways of violating the DRM associated with the artefacts. It finally outlines a security architecture to circumvent the threats.

2 COLLABORATIVE AI APPLICATION DEVELOPMENT

The purpose of data-driven AI is to analyse collected data in a smart way and come up with useful predictions about future data or provide new insights into existing data. However, in order to achieve good results, it is necessary to carefully prepare the data (*e. g.*, remove noise) and trim the algorithm parameters. The typical workflow model for data-driven AI, shown in Fig. 1 consists of five phases [7]:

i) ingestion,
ii) preparation (preprocessing)
iii) training (analytical processing),
iv) deployment
v) prediction

In the *ingestion* phase data is made available to the AI system either in the form of an existing dataset (*e. g.*, extracted from a data lake) or from live data (*e. g.*, collected from IoT devices).

The raw data must be converted to a format better suited for analysis. For example, data from heterogeneous sources that use different data type and syntaxes must be merged to a common format. This enables *feature extraction*, the retrieval of relevant attributes from data. Before data can be used for training it must be

ARES 2018, August 27–30, 2018, Hamburg, Germany

ACM ISBN 978-1-4503-6448-5/18/08…$15.00
https://doi.org/10.1145/3230833.3233268

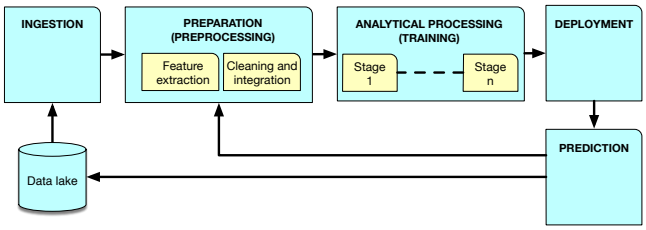

Figure 1: AI workflow model

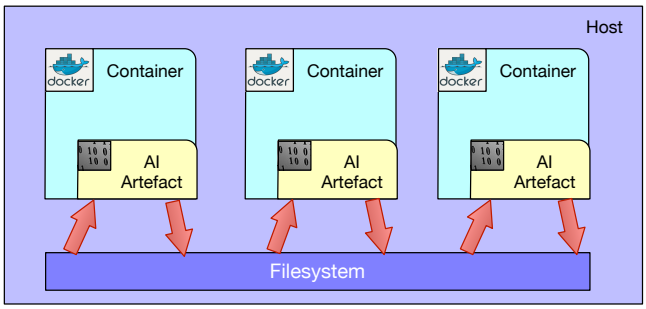

Figure 2: Original Bonseyes pipeline

cleansed from errors resulting from the data collection process [8]. Additionally, scaling and normalisation may be required to enable uniform handling of features expressed with different scales of reference. Similarly, data reduction and transformation can be used to reduce the size of the input data for the training phase. All these operations are performed in the *preparation phase*.

The *training phase* is where the actual analytical processing of the data takes place. The output from this phase can be either new insights about the analysed data (*e. g.*, the correlation between specific data items) or a model used for predictions.

A model needs to be *deployed* at a location where it can process input data. The location can vary between a powerful cloud computing environment to resource-constrained IoT devices, depending on the intended application.

The output of the models are predictions that can serve as input to business processes or decision-making systems. In some cases, a feedback loop is used, where the predictions are merged back into the original data lake or preparation phase to refine the existing model or train new models.

From a business perspective, we are witnessing the emergence of stakeholders that can provide access to high-quality data or algorithms. The co-dependency between data and algorithms in the AI workflow model suggests that collaboration between various stakeholders is required for developing complex, high-performant AI applications. To this end, the Bonseyes project is designing an AI marketplace that will enable such collaboration while maintaining privacy and enforcing DRM.

Bonseyes uses an Agile methodology for developing the marketplace. The current implementation of the AI workflow model is shown in Fig. 2 and is referred to as an *AI pipeline*. The light-blue rectangles in the figure are Docker containers. Inside the Docker

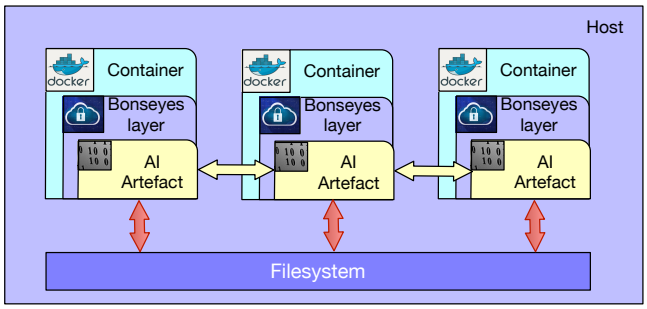

Figure 3: Improved Bonseyes pipeline

container resides an *AI artefact*. The artefact can be pure data, an interface to a data source or an algorithm used in the AI workflow.

The developers retrieve the required AI artefacts from a Docker repository and assemble the pipeline on their local system. The red arrows represent data transfers to and from storage (currently a regular filesystem). The transfers are needed to bring data in the pipeline and to save the output from intermediate stages and the final result. The data may include confidential information, hence the red color.

Fig. 3 shows the next implementation of the AI pipeline, where the AI artefact is wrapped inside a *Bonseyes Layer (BL)*. The purpose of the *BL* is to enable secure direct artefact communication without the need to store intermediate data. It also provides DRM mechanisms to control access to artefacts and enforce license management policies. The need for DRM mechanisms is driven both by business motives as well as legal requirements.

The new implementation of the AI pipeline supports also a distributed model, as shown in Fig. 4, which allows various components of the pipeline to be executed on different hosts. This model requires that the *BL* is extended to the hosts in order to facilitate workflow distribution and is the model considered for the remainder of the paper.

3 DRM AND PRIVACY REQUIREMENTS

This section aims to highlight the privacy requirements stemming from personal data regulations and describes the high level relationship between privacy and DRM. The impact of the regulation on the AI pipeline and the challenges for collaborative AI development is presented in Section 3.4.

3.1 GDPR privacy requirements

General Data Protection Regulation (GDPR) is the uniform approach towards all EU countries to provide the protection of a natural person while processing that individual's personal data. It will come in force in May 2018 as a replacement of European data protection Directive (EU Directive 95/46/EC) to give the EU citizens better control over their personal information. GDPR expands the notion of personal data to photos and audios, financial transactions, posts on social medias, device identifiers, location data, users login credentials, and browsing history, as well as genetic information. The new regulation applies in a wide territorial scope and it includes

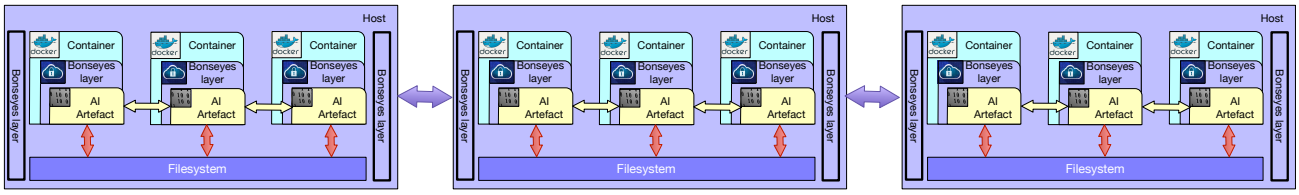

Figure 4: Distributed Bonseyes pipeline

all countries (EU or non-EU) that process the personal data of EU residents.

GDPR relies on six main principles for processing personal data namely a) lawfulness, fairness, and transparency; b) purpose limitation; c) data minimisation; d) accuracy; e) storage limitation; f) confidentiality and integrity [9].

GDPR limits the collection and storing of identification information of data subjects up to the minimum necessary required in order to safeguard data subjects' rights and freedom.

The new regulation expands the rights of data subjects and lists detailed obligations and responsibilities for two key entities: controllers and processors. The *controller*, who "determines the purposes and means of the processing of personal data", is responsible for implementing technical and organisational measurement to ensure the processing of data is performed as described in the regulation. The organisational measurement includes assigning a data protection officer, doing data protection impact assessment and risk mitigation plan. The technical measurement includes activities such as implementing accountability, pseudonymisation and data minimisation. A controller can delegate the processing of personal data to a *processor*, who "processes personal data on controller's behalf". It is therefore the responsibility of the controller to select processors who guarantee the implementation of appropriate technical and organisational measures that meet the regulation requirements and ensure the protection of data subjects [9].

3.2 GDPR & DRM

Digital rights management enable content owners to control the distribution and usage of their content. DRM systems conduct policies to manage the access to the digital contents. In general, DRM schemes for content protection are built on the assumption that the content owner has exclusive rights under the copyright law for the usage and distribution of the owned content. Consumer access to the content is regulated by a, typically non-transferable, license that is enforced through the DRM mechanism. GDPR does not attempt to challenge copyright law, but there are scenarios when the two can come into conflict. For example, if the protected content contains personal data collected from data subjects, then that content is subject to requirements implied by the GDPR. Another example consists of DRM mechanisms that use personal data to track the activity of individual users. The tracking data becomes also subject to the requirements of GDPR.

Typical DRM schemes can only allow or deny access to the entire content and not to portions of it. Furthermore, these schemes do not support fine-grain access control over user activities after the access is granted. Therefore, typical DRM schemes do not address the GDPR requirements (*e. g.* , recording processing activities). However, if DRM technologies are complemented with auditing and accounting capabilities, they can potentially help the GDPR controller to implement a technical measurement.

3.3 Privacy & DRM

Privacy is a fundamental human right [9] that can be defined in different ways. S. Warren and L. Brandeis proposed privacy as a "right to be left alone" [10]. At the same time, philosophers define the privacy as a right to limit access to individuals' information about lives, thoughts, and bodies [11]. These definitions show the broad concepts of privacy that apply to processing individuals' information and observing human behaviours and relations. The philosophical aspects of privacy is handled through information privacy law and data protection policies that regulate the process, storage, usage, and exchange of personal data. Some technologies, like DRM, seem to be aligned with privacy rights due to their basic natures. The main features of DRM, such as access control to protected content, restriction of unauthorised usage, and verification of the authenticity of identification data, are useful for implementing the legal requirements for privacy in a technical solution.

In [12], the authors proposed adaptations to DRM systems to address Privacy Rights Management (PRM). They define PRM as management of personal information according to the requirements of the EU Directive 95/46/EC [13]. To do that, the authors mapped the DRM entities to entities defined by the Directive. For instance, they mapped content owner to data subject, content distributor to controller, and user to processor. Unfortunately, their conclusion is that their method is not scalable in terms of issuing processing reports when the number of data subjects is very large. This remains a challenge in building DRM systems that support PRM-like features.

3.4 GDPR impact on AI pipelines

From a business point of view, the AI marketplace enables *artefact consumers* to obtain *licensed* access to AI artefacts owned by *artefact providers*. An entity can play both roles: it can be a provider for artefacts it owns, and it can consume artefacts from others. An artefact is coupled together with a license that specifies the terms of use for the artefact. Since the license is viewed as a legal document it must be anchored into existing laws and regulations (*e. g.* , GDPR) in order to be effective. A license can be encoded into a digital license to enable computer-based license management systems to monitor, detect and prevent license violations.

As described in Section 2, AI pipelines are setup in order to achieve specific goals such as to make predictions or obtain new insights from existing data. Since the purpose of the pipeline is

determined by the entity setting up, it is reasonable to conclude that the entity is acting as a controller in the GDPR sense and thus is responsible for complying with the GDPR. When viewed in the context of an AI marketplace, individual AI artefacts in the pipeline act as generic building blocks and should not be aware of the pipeline's higher purpose. It is thus reasonable to assume that artefact providers act as GDPR processors. However, some providers may implement the functionality of an artefact by chaining together several other artefacts, where each element of the chain consume the output of the previous element and provides input to the next. The determination of whether the provider of a chained artefact acts as controller or processor does not seem so clear cut in these cases.

The distinction between controllers and processors becomes even blurrier when the AI artefact contains a data source or pure data. The artefact provider could have collected and processed private data from its data subjects for specific purposes, thus being itself a controller according to GDPR. The provider may view the pipeline owner as a processor or they may establish a joint controller relationship [9].

This discussion highlights some of the immediate difficulties in attributing legal responsibilities to providers and consumers. It is not quite clear at the moment how an AI marketplace could *automatically* assign correctly (in a legal sense) the controller/processor roles to various entities participating in a pipeline. We believe that a tractable initial approach is to determine a baseline of requirements (legal and technical) for processors. All consumers and providers in the marketplace would be required to fulfill these constrains. This would enable controllers to delegate data processing to any processor.

4 DRM AND PRIVACY REQUIREMENTS FOR AI DEVELOPMENT

In general, DRM generates the license for each authorised entity in an AI pipeline. This section identifies general artefact usage constraints that can be embedded in a license:

i) **Validity** is defined by the allowed access duration to the artefact and is referred to as the license *validity period*. Additionally, the validity may also be constrained by the number of times the artefact can be executed, a so-called *n*-times use price model. Also, a license should be revokable if users breach the license agreement or to allow users to interrupt an automatic subscription renewal.
ii) **Beneficiary** constraint specifies the identity of the *license user*. An end-user license has a single user as beneficiary, whereas a group license allows an organisation to allocate and revoke licenses to its members according to own policies.
iii) **Purpose** constraint defines the *license scope* (*e. g.*, commercial, educational, or personal). This can even be tied to specific types of licenses (*e. g.*, GPL) that govern how derivative work (*e. g.*, AI applications) may itself be licensed. In addition, legislation, such as the GDPR, may stipulate how data must be collected, stored, shared and processed. The usage purpose constraint must be able to incorporate this type of law-derived policies.
iv) **Location** constraints define where the artefact can be utilised. This constraint can be either topological or geographic. The topological location defines the networked hosts that are allowed to use the artefact. For the Bonseyes project, the simplest topological constraint restrict artefact usage to the set of Bonseyes authorised hosts. However, artefacts can also be constrained to *virtual premises*, which are smaller subsets of Bonseyes authorised hosts[14]. The geographical location defines where in the world the artefact can be used or is prohibited from being used. This allows hosts and artefact to comply with local laws and regulations, such as EU Data Protection Directive and its successor, GDPR.
v) **Peering** constraint regulates which entities the artefact is allowed to interact with. The peers can be the successor or predecessor artefact in a pipeline or entities that monitor and control the operation of the artefact.

A license management system must protect the interests of both artefact consumers and providers, according to license constraints. This means that consumers are not prevented from using the artefact while the license is valid, but also that they cannot continue to use it if the license expires, is revoked for a legitimate reason, or if the artefact is used for a purpose or at a location prohibited by the license. Similarly, the providers must be prevented from blocking access to an artefact for consumers with a valid license, while at the same time retain the possibility to revoke a license when its terms are breached. The collaborative nature of the AI marketplace raises some interesting challenges in satisfying these requirements simultaneously.

To ease the threat analysis, we will consider two classes of misbehaving users: regular users and malicious users. Regular users can be consumers that try using the artefact in conflict to the license constraints, or providers that attempt to prevent consumers from using an artefact although a valid license exists. Common for this case is that the users are either unaware that they are breaching the license agreement, or that they are aware, but do not employ any advanced means (*e. g.*, reverse engineering or code injection) to achieve their goals. Therefore, we consider these potential attacks as *simple threats*. On the other hand, the malicious users are assumed to be skilled attackers that may insert exploits into the AI pipeline or instrument the artefact's hosts in order to bypass the license and obtain unfettered access to the artefact of interest. We consider these to be *advanced threats*. Provider attempts to fraudulently prevent consumer with valid licenses from using an artefact belong also to the advanced threats category.

4.1 Simple threats

We begin by considering two obvious threats:

T-1: A user can modify license contents to bypass usage constraints
T-2: The AI marketplace repudiates the issuance of a license

It is assumed that the license contents are stored inside the BL that encapsulates the AI artefact. An additional assumption is that the license contents as well as the AI artefact are digitally signed by the marketplace. The signature enables the BL to detected fraudulent changes to the license contents and asserts the origin of the data thus preventing the marketplace from repudiating an issued license. We assume the signature algorithm itself (*e. g.*, RCA-PSS, DSA or ECDSA) when used with a reasonable key length is infeasible to break. This counteracts threat T-1 and T-2.

A *license validity period* begins on a specific start date and stops at an expiration date. The expiration date can be left undefined by the licensor if perpetual validity is desired. Additionally, if an *n*-times use pricing model is used, the validity can be further restricted by the number of times the artefact is executed. A license can be revoked before the expiration date for reasons mentioned previously in this paper. To determine if a license is valid, the BL is dependent on having access to a time source, such as a Network Time Protocol (NTP) server, and to a license revocation database. We consider threats against the integrity or availability of the time source and the license revocation database. More specifically, we require security mechanisms are put in place to protect against:

T-3: Blocking communication with the time source
T-4: Blocking communication with the license revocation server
T-5: Spoofing a time source
T-6: Rolling back the time on the artefact host to use the artefact past expiration date

Threat T-3 and T-4 are essentially DoS attacks on network services required by the license management system. To counteract them, the BL should refuse to execute the artefact unless communication with the servers is possible. The downside of this approach is that AI pipelines become unavailable during periods of none-malicious network outage (*e. g.*, on mobile units performing handoffs). This situation can be improved by introducing *grace time*, which is an acceptable duration for communication outage. Threat T-5 can be handled by using NTP Autokey [15] or a time server authentication mechanism similar to it. To defend against threat T-6, we assume that after an initial connection to the time source, the artefact is able to detect deviations from monotonically increasing time.

The *license beneficiary* uniquely identifies the licensed consumer. This information is determined at the time the license is acquired. Its presence in the license enables accounting and usage tracking. Since the license and AI artefact are protected by a digital signature, it is assumed that no simple threats exists against the integrity of this information. However, it is important to protect against the execution of an artefact by an unlicensed user (*e. g.*, in the case a pirate copy is made).

T-7: Artefact execution by unlicensed user

To counteract T-7, it is necessary to require users to authenticate themselves to the BL with a digital user certificate signed by a Bonseyes certification authority (CA). First, the BL would have to establish a chain of trust to the root CA and then verify the validity of the certificate by consulting certificate revocation lists (CRLs) or through invoking an Online Certificate Status Protocol (OCSP) responder. If the certificate is valid, the BL can verify if the license beneficiary matches the distinguished name (DN) in the user certificate. Finally, the BL would present a challenge to the user agent to ascertain the user in possession of the private key associated with the certificate. It is important to realise that this defence will not work against attackers that are able to obtain the licensed artefact, the signed certificate and the private key.

License purpose is metadata encoding the permitted use scope for the artefact. Hosts in a virtual premise have similar metadata configured for their BL. The artefact BL and the host BL each expose their use scope. Thus, the artefact can decide if a specific host meets its purpose constraint and the host can decide if it can allow artefact execution. For example, an educational virtual premise may refuse to execute commercial artefacts, or an artefact can avoid running on hosts that does provide adequate privacy protection (*e. g.*, according to GDPR). This constraint faces the following threats:

T-8: Host breach against license purpose
T-9: Artefact breach against license purpose

In the case of T-8, the host is configured to present a false use scope that entices the artefact to execute its payload. This threat can be addressed by requiring that before an organisation can deploy a Bonsyes host it must obtain a digitally signed license for that host from the Bonseyes CA. The license carries the approved use scope and is tied to the host's hardware through a hardware ID [16]. The BL in the host can recompute the hardware ID and verify that it matches the one covered by the license, before presenting the host license to the artefact BL. Threat T-9 considers the situation where the artefact is attempting to illegitimately obtain execution privilege on the host. In this case, the host BL must verify the artefact license to ensure its use case is allowed.

We foresee the following topological location threats:

T-10: Artefact execution outside a virtual premise (*i. e.*, host lacking a valid BL)
T-11: Artefact execution on wrong virtual premise

These threats are similar to the T-8, but they are more generic in scope. Virtual premises are host pools for AI pipelines with different constraints. The license purpose is an example of such a constraint. Performance can be another constraint. For example, one may define a development premise as a pool of workstations used by data scientists for research and development work. At the same time, a production premise can consist of powerful servers used for high-performance AI pipelines. Threat T-10 describes the trivial case where a host lacks the BL or contains an invalid BL, thus being outside any virtual premise. If the host BL is missing, the artefact BL cannot establish a connection to it and must abort execution immediately. An invalid BL is either old deprecated code (detectable through a revoked license) or contains an invalid license (*e. g.*, expired or missing a valid trust chain). Also in this case the artefact BL must abort execution.

Threat T-11 considers the situation where an artefact is restricted to a specific virtual premise and by mistake or intentionally is executed on a different premise. A scenario illustrating this threat is that of an artefact under development (*e. g.*, using an untested algorithm or containing low-quality data) being mistakenly deployed on a production premise, thus degrading the quality of the results. To counteract this threat we propose that host and artefact licenses contain a set of virtual premise identifiers that must match in order to allow artefact execution.

T-12: Artefact execution outside allowable geographic region.

Threat T-12 is a special version of T-11. In this case, we consider mobile virtual premises, a host or a set of hosts that can move or be transported geographically. The issue we try to capture is that when geographic boundaries are being crossed (like country or state borders) different jurisdiction may apply to computation and data. For example, some countries may have laws that severely curtail data privacy and where certain types of computation (*e. g.*,

cryptography) can be considered a criminal offence. Theoretically, this threat can be dealt with in a similar manner as with T-10 and T-11, by encoding allowed geographic regions in the license. Unfortunately, enforcing this type of constraint is quite difficult. The reasons are firstly, that there is no single method that can do accurate Internet geolocation without assistance from GPS or cellular networks and, secondly, evasion methods through proxies and anonymization networks are quite successfull [17]. Therefore, more research is required to determine efficient countermeasures for this threat.

T-13: Unauthorized peering.

In an AI pipeline there are multiple artefacts chained together. Threat T-13 is about scenarios where an artefact in a chain must determine if the predecessor and the successor artefact are authentic and part of the same virtual premise. It applies also to the communication between control entities and artefacts. This threat can be addressed by having peering entities exchange licenses with each other. The neighbour licenses are verified and each peer is required to demonstrate that it has possession of the corresponding private key.

4.2 Advanced threats

In this section we consider threats from skilled malicious users. The assumption is that this type of users are able to modify the code for both the artefact BL as well as the host BL. Also, they are legitimate AI marketplace users and can, for example, obtain valid host licenses. Thus, these attackers have the capability to circumvent any authorization and authentication mechanisms running on systems that are under their control. Consequently, the defence mechanism for T-1 to T-13 are defeated on all virtual premises under the control of the attacker. More specifically, it means that license management mechanisms can be bypassed and AI artefacts obtained from the marketplace cannot impose any limitation on how they are used. The attacker can also extract the actual algorithms or data encapsulated by the artefact and make it available outside the marketplace. Even if artefacts employ encryption as a copy-protection scheme, they cannot be protected against this type of attacks. The crux of the problem is that for an artefact to be useful, it needs to be decrypted at some point. The attackers can instrument the hosts under their control using a tool such as Intel PIN[1] and locate the decrypted artefact payload. This type of approach was successfully used to defeat movie DRM [18].

Although the outlook for defending against advanced threats looks quite gloom, there are two complementary approaches that can raise the difficulty of mounting a successful attack. The first approach aims at providing a trusted computing environment for license management mechanisms to execute, in particular one in which the integrity of host and artefact BL are protected. The ideas about trusted computing have been vastly explored in the past and have lead to the design of the Trusted Platform Module (TPM), which is an international standard for a secure crypto-processor. A discussion about TPM is outside the scope of this paper and the interested reader is referred to [19] and [20]. The TPM is a passive chip located on the computer mainboard and can perform measurements (*i. e.*, compute cryptographic hash values) on software and its dependencies, before the software is executed. In addition, it enables a task called *remote attestation* where the measurements of a host are communicated over a secure cryptographic channel to a remote attestation server. The attestation server can compare the measurements with a set of pristine values collected in advance and, if they match, authorize the host to engage in activities with the rest of the system.

The TPM has been plagued from the start by a number of deficiencies and major manufacturers such as Intel and AMD have worked over the years to address them. Intel has recently released the Software Guard Extensions (SGX) for their CPUs, bringing support for secure *enclaves*. Applications and data running inside enclaves are protected from other software running on the same platform. Most recently, this mechanism has been used to protect Docker containers[21]. The downside of TPM- or SGX-based solutions is that the actual hardware module (or processor extension in case of SGX) is not available on all platforms. It should be noted that although SGX, if correctly used, can provide considerable protection against tampering with the computing environment, there are already attacks against it that are considered practical [22].

Our second proposed approach for providing a trusted environment is to remove the hosts from the control of a potentially malicious users. In particular, we envision a scheme, similar to [23], where the actual implementation of the AI pipeline is done by a third party, such as a cloud provider. The assumption is that the cloud provider's main business interest is in selling computation and storage and thus the provider has no incentive to engage in malicious activities agains the AI marketplace and its users. Artefact consumers interface with the artefact through well-defined interfaces, without having direct access to the artefacts or the hosts executing artefacts. Although, this approach can provide a high level of security against malicious users, it requires a major readjustment to current AI development practices and is thus regarded as a major inconvenience by the developers.

5 AI MARKETPLACE SECURITY ARCHITECTURE

We propose three architecture models based on the requirements described in the sections 3 and 4 . The single and multi-host architectures are suitable to address the simple threats and the 3rd party cloud architecture aims at mitigating the advanced threats. The main entities in our protection schemes are as follows:

i) **MarketPlace (MP)**: it is the interaction point between all parties. The MP enables collaboration between the main entities listed below.
ii) **Artefact Provider (AP)**: makes the AI artefacts available by registering them in the MP. AP is the owner of the artefact having rights to describe the artefacts' access policy, license type, and privacy restrictions.
iii) **Security Manager (SM)**: is the heart of the license management system. It generates the access policy per artefact dynamically based on the AP description. SM has the multiple functionalities such as the DRM enforcement system, remote attestation, AAA, and certificate management. It is responsible for initialising the artefact per request. It manages the system log, license, privacy policies and the execution of each artefact.

[1] https://software.intel.com/en-us/articles/pin-a-dynamic-binary-instrumentation-tool

iv) **Data Scientist (DS)**: is the entity that needs the AI artefact for testing or developing new applications. It is regarded as the artefact consumer.
v) **Computation Center (CC)**: is the datacenter which provides the infrastructure for executing the AI artefact. It can be located on the DS premise or with a 3rd party cloud provider and is required to establish trust to SM *e. g.*, through trusted computing. Therefore, the CC hosts should be equipped with Trusted Platform Modules (TPMs) or equivalent.
vi) **Virtual Premise (VP)**: spans the computational capabilities of the hosts involved and provides the trusted execution environment to protect the AI artefact. It prohibits the communication between the artefact and outside world. The MP and SM are part of every VP.

We assume the SM is under the control of the MP and helps the secure collaboration between parties. The payment transaction is beyond the scope of this paper. The BL contains the privacy rights and access policies and it is customised by the SM for each artefact instance and host.

5.1 Artefact retrieval and license enforcement

All entities must be authenticated to the MP before requesting access to the AI artefact. After successful payment procedure, the SM generates the license for the user based on the service term agreements between the DS, AP, and the MP. The SM provides customised privacy rights and inserts their description inside the BL. The DS could download his own copy of the artefact to execute it in his own premise if the requirements are met. The BL establishes a secure connection to the SM to check the validity of the license in order to permit artefact execution inside the DS premise. Blocking the communication channel to the SM is considered a policy violation and the BL will halt in that case the execution of the artefact. If the communication channel is unblocked and the artefact and the DS have valid licenses, the artefact will begin execution and a log entry will be submitted to the SM. If a policy violation occurs during runtime, the BL will log to the SM and exit artefact from execution mode.

The DS does not have direct access to the artefact, but instead must communicate with artefact through predefined interfaces denoted by blue arrows in Fig. 5- 7. The interfaces are typically managed through an integrated development environment (IDE) such as Eclipse, denoted by DS Workbench (staging area) in the figures.

The general activities to obtain an artefact from the MP are the same for different architectures and the exact artefact retrieval protocol is under development.

5.2 Single Host Architecture

This model is adapted to current AI development practices, where the DS has access to the entire pipeline in his own workstation. It provides minimum security compared to the other two architectures and it is targeted towards a convenient work mode for developers. In addition, it provides convenient artefact mobility. The artefact could move either together with the host (*e. g.*, laptop) where it is installed, or the artefact itself can be transferred to other BL-enabled host (*e. g.*, being carried on a USB memory stick). This architecture prefers simplicity over security and thus does not offer any protection against advanced threats.

The architecture depicted in Fig. 5 shows the pipeline in a single authorised host. The yellow arrows represent the communication through BL interfaces between peering artefacts, whereas the blue arrows denote the communication between the interfaces in the DS workbench and the actual artefacts. The green arrow represents the communication channel between BL entities and the SM that was described in Section 5.1. The DS follows the general approach in Section 5.1 to obtain the artefacts from the MP, assemble and execute the pipeline in the hosting workstation. The DS can move artefacts to a different topological and/or geographical location if this is not prohibited by the artefact policies.

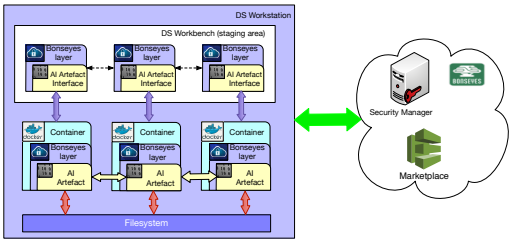

Figure 5: Isolated pipeline host

5.3 Multi-Host Architecture

The multi-host architecture depicted in Fig. 6 addresses the situation when the computing requirements of the pipeline exceed those available on a single host. In this model, the DS can extend its pipeline to multiple hosts. The containers shown in grey colour represent artefacts belonging to a different pipeline. The green arrow is a communication channel to the SM. Each host belonging to a pipeline must establish its own secure channel to the SM. A SM must issue licenses for multiple authorised hosts under the control of the DS. This architecture is designed to improve the performance of the single host architecture by facilitating the implementation of the pipeline across several hosts.

The general approach described in Section 5.1 is used to receive the artefact. However, the hosts are under the control of the DS and thus, when the DS is malicious, this architecture is still vulnerable to the advanced threats described in Section 4.2.

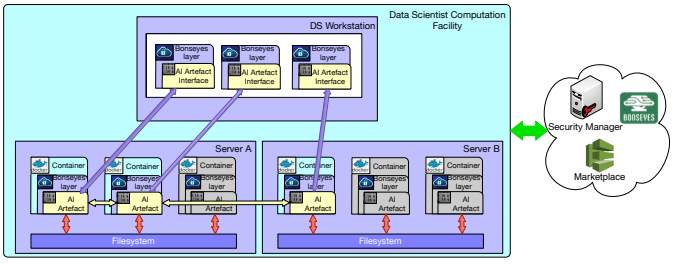

Figure 6: AI pipeline execution at a CC hosted by the DS

5.4 3rd Party Cloud Architecture

Fig. 7 shows the 3rd party cloud architecture aiming to address the advanced threats. Following the approach from Section 5.1, the DS authenticates to the MP and requests for the AI artefact to be executed in a 3rd party CC. The 3rd party CC must be trusted by the MP and be authenticated against the SM. In addition, the provider must install in advance the BL on hosts available for BL pipelines. The CC must be out of the control of the DS and moreover have no interest in the artefacts. More general, we assume there is no collusion between the CC and the DS. The procedure from Section 5.1 will follow and the provisioned artefacts will instantiate for execution in the 3rd party CC. The BL in the artefact must be able to communicate with the SM through secure channel to enable license management. The BL permits the artefact execution if the communication channel is alive and the license requirements are matched.

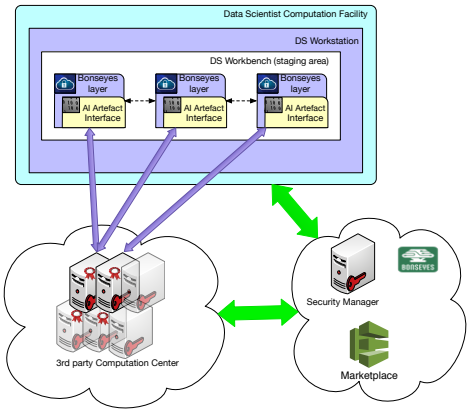

Figure 7: AI pipeline execution at a CC hosted by the 3rd party cloud provider

The green arrows in Fig. 7 represent the communication channel between the SM and all other participating entities. To prevent compromised artefacts belonging to malicious DSs from exfiltrating entire datasets, the total volume of data retrieved as output of the pipeline is restricted . The limitation is both in terms of request rates for pipeline execution as well as on the size of the output for individual requests. We believe this architecture is more secure against the advanced threats because artefact consumers are forced to use dedicated interfaces instead of having direct access to raw artefacts or the executing hosts. This measure raises the difficulty to circumvent DRM mechanisms by compromising pipelines with instrumented artefacts and hosts.

6 CONCLUSION

The paper describes the privacy requirements for enabling collaborative concepts in AI application development. It investigates the ordinary DRM constraints to address privacy requirements by regulations and proposes a DRM scheme to achieve fine-grain access to the artefacts. This work describes the possible threats in collaborative AI and suggests three architecture models to meet the requirements. The paper aims at a practical solution for the privacy challenge in AI application development and at facilitating collaboration between different stakeholders.

In the future work, we will implement the proposed architecture models and investigate the vulnerability of the models based on the identified threats.

Acknowledgment: This project has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No 732204 (Bonseyes). This work is supported by the Swiss State Secretariat for Education Research and Innovation (SERI) under contract number 16.0159. The opinions expressed and arguments employed herein do not necessarily reflect the official views of these funding bodies.

REFERENCES

[1] I. Stoica, D. Song, R. Ada Popa, D. Patterson, M. W Mahoney, R. Katz, A. D Joseph, M. Jordan, J. M Hellerstein, J. E Gonzalez, et al. A berkeley view of systems challenges for AI. *arXiv preprint arXiv:1712.05855*, 2017.
[2] T. Llewellynn, M Fernández-Carrobles, O. Deniz, S. Fricker, A. Storkey, N. Pazos, G. Velikic, K. Leufgen, R. Dahyot, S. Koller, et al. BONSEYES: platform for open development of systems of artificial intelligence. In *Proceedings of the Computing Frontiers Conference*, pages 299–304. ACM, 2017.
[3] PipelineAI - Home. available at https://pipeline.ai/.
[4] Orange Data Mining Fruitful & Fun. available at https://orange.biolab.si/.
[5] Data Market Austria. available at https://datamarket.at/.
[6] Industrial Data Space e.V. available at http://www.industrialdataspace.org/.
[7] Serdar Yegulalp. Data in, intelligence out: Machine learning pipelines demystified. InfoWorld, May 2017.
[8] Charu C. Aggarwal. *Data Mining: The Textbook*. Springer International Publishing Switzerland, 2015. ISBN: 973-3-319-38116-9.
[9] European Parliament and Council of the European Union. Regulation (EU) 2016/679 of the European Parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/ec (general data protection regulation). In *Official Journal of the European Union*, volume 59. May 2016. ISSN: 1977-0677.
[10] S. Warren and L. Brandeis. The right to privacy. *Harvard Law Review*, 4(5):193–220, December 1890.
[11] Ferdinand David Schoeman, editor. *Philosophical Dimensions of Privacy: An Anthology*. Cambridge University Press, Cambridge Cambridgeshire ; New York, 1st edition, November 1984.
[12] Steve Kenny and Larry Korba. Applying digital rights management systems to privacy rights management. *Computers & Security*, 21(7):648–664, November 2002.
[13] Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data, November 1995.
[14] V. Mehri and K. Tutschku. Flexible privacy and high trust in the next generation internet: The use case of a cloud-based marketplace for AI. In *Proceedings of SNCNW*, Halmstad University, Sweden, 2017.
[15] Brian Haberman and David L. and Mills. *RFC 5906: Network Time Protocol Version 4: Autokey Specification*. IETF, June 2010.
[16] Microsoft. Specifying hardware IDs for a computer. Available at https://docs.microsoft.com/en-us/windows-hardware/drivers/install/specifying-hardware-ids-for-a-computer, 2017.
[17] James A. Muir and Paul C. Van Oorschot. Internet geolocation: Evasion and counterevasion. *ACM Computing Surveys*, 42(1), December 2009.
[18] R. Wang, Y. Shoshitaishvili, C. Kruegel, and G. Vigna. Steal This Movie: Automatically Bypassing DRM Protection in Streaming Media Services. In *Proceedings of 22nd USENIX Security Symposium*, pages 687–702, Washington DC, USA, 2013.
[19] Keylene Hall, Tom Lendacky, Emily Ratliff, and Kent Yoder. Trusted computing and linux. In *Proceedings of the Linux Symposium*, Ottawa, Ontario, Canada, 2005.
[20] Bryan Parno, Jonathan M. McCune, and Adrian Perrig. Bootstrapping trust in commodity computers. In *Proceedings of IEEE Symposium on Security and Privacy*, Oakland, CA, USA, 2010.
[21] S. Arnautov, B. Trach, F. Gregor, T. Knauth, A. Martin, C. Priebe, J. Lind, D. Muthukumaran, D. O'Keeffe, ML. Stillwell, Goltzsche. D., D. Eyers, R. Kapitza, P. Pietzuch, and C. Fetzer. SCONE: Secure linux containers with intel SGX. In *Proceedings of USENIX OSDI*, Savannah, GA, USA, 2016.
[22] Ferdinand Brasser, Urs Müller, Alexandra Dmitrienko, Kari Kostiainen, Srdjan Capkun, and Ahmad-Reza Sadeghi. Software Grand Exposure: SGX Cache Attacks Are Practical. *arXiv:1702.07521 [cs]*, February 2017. arXiv: 1702.07521.
[23] Ronald Petrlic. Privacy-preserving digital rights management in a trusted cloud environment. In *Proceedings of TrustCom*, Liverpool, England, UK, 2012.