Chapter 9
Building Trust in Data Spaces

Monika Huber, Sascha Wessel, Gerd Brost, and Nadja Menz

Abstract Data is becoming increasingly valuable and must be protected. At the same time, data becomes an economic asset and companies can benefit from exchanging data with each other. The International Data Spaces enable companies to share data while ensuring data sovereignty and security.

Data providers can keep control over the processing of their data by utilizing usage control policies, including the verification that these usage control policies are enforced by the data consumer. For this, data processing devices, called connectors, must prove their identity and the integrity of their software stack and state.

In this chapter, we present the overall security concept for building trust in data spaces enabling data sovereignty and usage control enforcement. The concept builds on a certification process for components and operational environments utilizing the multiple eye principle. This process is technically mapped to a public key infrastructure providing digital certificates for connector identities and software signing. Finally, the third building block is the architecture and system security of the connectors where usage control must be enforced, the identity and integrity of other connectors and their software stack and state must be verified, and the actual data processing happens.

9.1 Introduction

Data is an important asset for many companies. In particular, data collected during manufacturing or usage of their products could provide insights in business secrets. Such data is therefore often carefully protected and not shared with others. On the

M. Huber (✉) · S. Wessel · G. Brost
Fraunhofer AISEC, Garching, Germany
e-mail: monika.huber@aisec.fraunhofer.de; sascha.wessel@aisec.fraunhofer.de; gerd.brost@aisec.fraunhofer.de

N. Menz
Fraunhofer FOKUS, Berlin, Germany
e-mail: nadja.menz@fokus.fraunhofer.de

B. Otto et al. (eds.), *Designing Data Spaces*,
https://doi.org/10.1007/978-3-030-93975-5_9

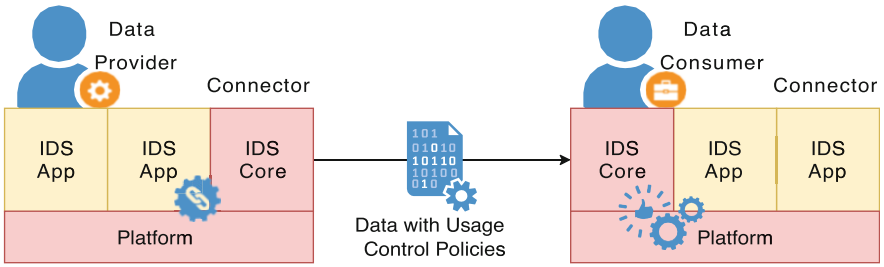

Fig. 9.1 Data exchange in the International Data Space

other hand, companies can benefit from exchanging data with each other and processing combined data sets as a result. Due to the importance of the confidentiality of the data, such a data exchange requires a high trust level between the communication partners.

For that purpose, the International Data Spaces (IDS) enable companies to share data while ensuring data sovereignty and security. A typical scenario for the data exchange in the IDS is depicted in Fig. 9.1. A data provider offers data that is then shared with a data consumer. The data exchange and processing of data is conducted by IDS connectors on both sides of the communication. Further details on this scenario are provided in Sect. 9.2.

In the following, we introduce how trust is established in the IDS ecosystem with the focus on the communication between data providers and data consumers. One key value of the IDS is the possibility for data providers to keep control over the processing of their data by utilizing usage control policies. Data providers can verify that their usage control policies are enforced by their communication partners. For that purpose, connectors can prove their identity and the integrity of their software stack and state. Data consumers can utilize the same mechanisms to ensure the trustworthiness of the data provider. For this, the following three aspects are required in the IDS:

- *Certification Process*: Processes providing verified information concerning the utilized components and involved companies based on an evaluation conducted by independent trusted third parties.
- *Connector Identities and Software Signing*: Mechanisms to technically represent the results from the certification process in order to make the identity and integrity of components verifiable.
- *Connector System Security*: Security of the connector to fulfill the requirements of the certification and to properly conduct the verification of the component's identity and integrity.

In the following, we first describe the overall concept in more detail in Sect. 9.2. Afterward, we explain the certification process in Sect. 9.3, the utilized identity infrastructure and trust model in Sect. 9.4, and the connector system security in Sect. 9.5. Finally, we conclude in Sect. 9.6.

9.2 Data Sovereignty and Usage Control

In the following, we first provide details regarding the communication between data providers and consumers in Sect. 9.2.1. Based on the attacker model described in Sect. 9.2.2, we afterward describe the building blocks for establishing trust and security in the IDS in Sect. 9.2.3.

9.2.1 Data Provider and Data Consumer

One goal of data sovereignty is to allow data owners to control the usage and processing of their data. The IDS offer a solution for companies to keep their data sovereignty even after transferring data to others. The data providers can define usage control policies which the data consumers will have to fulfill. Examples for such policies are the restriction of processing to defined time periods or countries as well as requiring anonymization of data before further processing.

In order to technically implement usage control, both communication partners must implement a trustworthy system that supports the definition of usage control policies (data provider) and the verifiable enforcement of those policies (data consumer). Those systems are called connectors in the IDS.

Figure 9.1 depicts a minimal scenario for a communication in the IDS where one data provider wants to transfer data to one data consumer using IDS connectors. Every connector consists of a Platform providing a runtime to run data processing IDS Apps and implementing system security mechanisms, as well as one dedicated IDS Core App implementing core features for secure connector-to-connector communication. The Platform, the IDS Core, and one data processing IDS App have full access to the confidential data and must therefore be trusted. More details on connector architectures are described in Sect. 9.5.1. The blue gears in Fig. 9.1 symbolize the usage control policies, which are defined by the data provider on the left side, transmitted to the data consumer, and enforced by the data consumer's connector on the right side.

9.2.2 Protection Goals and Attacker Model

A connector must guarantee data confidentiality and the integrity of the connector stack, which implements the enforcement of usage control policies, as its primary *protection goals*. The secondary protection goal is the integrity of the data. Availability is only ranked third, as within the IDS confidentiality is always prioritized higher than availability.

Depending on the business case and the type and amount of data, the data itself can be of different value. Therefore, the IDS consider different attacker models, and

each company can decide against which type of attacker the protection goals must be defended.

In the IDS the following two classes of attackers are addressed:

- *Remote attacker*: The attacker can eavesdrop and modify communication between connectors. The attacker cannot break cryptography if it is state of the art. The attacker does not have access to the connector.
- *Local attacker*: In addition to the previous attacker class, the attacker has full control over the connector. This includes the capability to access unprotected data on the connector or modify unprotected software on the connector. However, the attacker cannot break state-of-the-art cryptography and cannot own the hardware-based root of trust for measurement. The attacker does not have access to private keys, e.g., used for decryption or authentication.

9.2.3 Building Blocks

To reach the protection goals defined in Sect. 9.2.2, the IDS is built on the following three modules depicted in Fig. 9.2 which must fit together and are described in detail in the following sections:

- Section 9.3—Certification Process: The *Certification Process* implements the evaluation of participating companies and connectors utilizing a multiple eye principle to gain verified information regarding the functionalities and security of connectors and companies. In order to address different use cases, the IDS certification offers different security and assurance levels for both types of certification.
- Section 9.4—Connector Identities and Software Signing: In order to technically represent the *Certification Process* described above, the IDS define a Public Key Infrastructure (PKI) for managing the identities of persons (users) as well as devices in the IDS. The user certificates are used to sign the results from the

Fig. 9.2 Building blocks for building trust in data spaces

certification process in the form of company descriptions and software manifests. Together with the device certificates, those signed descriptions are used to prove the identity and integrity of the IDS connectors.

- Section 9.5—Connector System Security: The *Certification Process* defines numerous security requirements for IDS connectors. Depending on the architecture of the connector, different components might have access to the data. All these components must fulfill these requirements and must be evaluated by trusted third parties. In order to ensure that a communication partner really utilizes those certified software components, the IDS offer an attestation protocol for remote identity and integrity verification. This protocol uses the device certificates, company descriptions, and software manifests introduced before.

9.3 Certification Process

The IDS utilize a certification scheme [1] to get confirmed and verifiable information on the connectors and the companies operating them. The underlying principle for this certification is described in Sect. 9.3.1. Afterward, we provide details concerning the *Component Certification* focusing on the connector implementations in Sect. 9.3.2 and the *Operational Environment Certification* for the companies operating IDS components in Sect. 9.3.3.

9.3.1 Multiple Eye Principle

All connectors as well as their operators must provide a sufficient level of (data) security and must correctly implement the IDS standards. Participants need to be able to trust that other participants previously unknown to them truly fulfill these requirements. Therefore, the IDS utilize a certification scheme [1] which ensures that the needed information concerning the communication partners is verified by multiple independent parties. The procedure for certification is depicted in Fig. 9.3 with

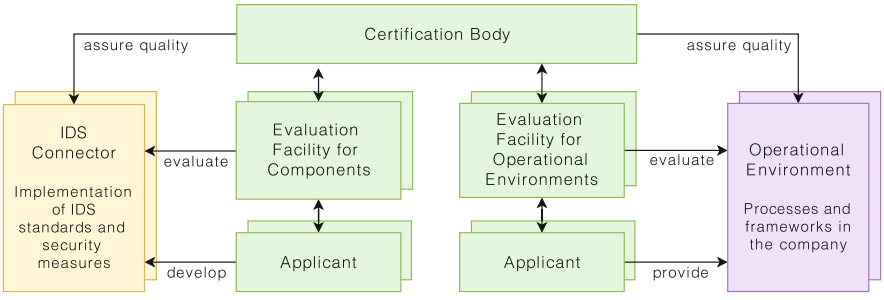

Fig. 9.3 Certification process for components and operational environments

the left side showing the certification of IDS connectors and the right side illustrating the operational environment certification. As the figure shows, the three involved parties and the procedures for the two different certification types are equivalent. The roles of the depicted entities and their responsibilities in the certification process are described in the following:

- The *applicant* is the person or organization that starts the certification process by contracting an evaluation facility and providing it with all necessary information and evidence for conducting the evaluation. For components this role is typically filled by the developer and for operational environments by (a representative of) the company that wants to operate an IDS component.
- An *evaluation facility* is responsible for assessing whether a connector or operational environment fulfills the specified requirements for the intended security level. The evaluators perform a thorough evaluation of the provided information following a standardized evaluation procedure. As a result, they provide an evaluation report to the certification body which describes the conducted evaluation work and its results. The concrete depth and scope of the evaluation depend on the desired level of security and assurance as described in Sects. 9.3.2 and 9.3.3.
- The IDS *certification body* is responsible for ensuring the comparability and quality of all conducted evaluations. For that purpose, the certification body must assess the competence of evaluation facilities and approve them before they are allowed to conduct evaluations in the IDS. For each conducted evaluation, the certification body reviews the provided evaluation report to ensure the correct execution of the evaluation. The certification body makes the final decision about the award or denial of each certificate.

9.3.2 Component Certification

Connectors play an essential role in the IDS. They are responsible for executing the communication and data processing in the IDS and must therefore fulfill many requirements concerning functionality and security. Consequently, the certification of IDS connectors focuses on interoperability and security. In order to address the different use cases in the IDS, the certification is designed to offer different security profiles and assurance levels as depicted in Fig. 9.4.

The three security profiles define an increasing set of security measures and requirements:

- The *Base Security Profile* defines a set of minimal requirements for participating in the IDS which focus on protection against a remote attacker. It allows companies to try out the IDS and can be used for handling open data.
- The *Trust Security Profile* defines more thorough security requirements to be fulfilled. The goal is to protect data from a remote attacker and provide all necessary means to realize usage control on the connector. However, for local

attack scenarios the profile only aims to prevent accidental misuse from an administrator but does not consider the administrator as a potential attacker.
- The *Trust + Security Profile* fills this gap. It has almost equivalent (security) requirements as the *Trust Security Profile*, but in addition aims to keep all the required capabilities and assurances against a powerful local attacker such as a malicious administrator.

The criteria catalog for these three security profiles [2] combines IDS-specific functional requirements aiming to achieve conformity with the IDS Reference Architecture Model [3], requirements from the industrial security standard IEC 62443-4-2 [4], and requirements for secure software development.

In addition to security profiles, the IDS certification offers three assurance levels that define the depth of the evaluation:

- As a low access barrier, the IDS certification allows a *Checklist Approach* as an assurance level for the *Base Security Profile*. The applicant fills out a questionnaire covering all applicable requirements and an automated test suite is used to run a set of interoperability and security tests. No evaluation facility is involved, but the questionnaire and the result from the automated test suite are reviewed by the IDS Certification Body.
- For a *Concept Review*, the applicant is required to provide the evaluation facility with detailed documentation for the connector as well as a working instance of the connector implementation. The evaluation facility conducts a review of the provided concepts and security measures as well as practical testing concerning the functionality and security of the implementation.
- The *High Assurance Evaluation* builds on the *Concept Review* evaluation and extends it with in-depth source code reviews and an on-site visit to the development site of the connector.

The described assurance levels and security profiles can be combined into six possible ways as depicted in Fig. 9.4. Whenever applicants want to get a connector certified, they can decide which security profile and assurance level they want to

Fig. 9.4 Connector certification levels

target. In reverse, each IDS participant can decide which expectation it has on the certification of its communication partner's connector as detailed in Sect. 9.5.2.

9.3.3 *Operational Environment Certification*

In addition to the certification of technical components, the IDS certification scheme [1] defines an operational environment certification that is required for each company offering connectors in the IDS. It requires companies to fulfill defined standards for management processes and infrastructure in order to ensure a secure operation of IDS components. Analogous to the connector certification, the operational environment certification is offered with different security and assurance levels as depicted in Fig. 9.5.

Depending on the role a company wants to fulfill in the IDS, there are different security levels with increasing requirements:

- The *Entry Level* is meant for companies that want to give the IDS ecosystem a try and gain access with minimal effort. Thus, this level only includes minimal requirements for the procedures and security mechanisms in the company.
- The *Member Level* defines the standard requirements that are meant to be met by all companies participating in the IDS. This level is recommended for data providers and data consumers introduced in Sect. 9.1.
- The *Central Level* is designed for companies that take on specific trust-building tasks in the IDS. An example is the issuing of identity certificates described in Sect. 9.4.1. These companies must meet a more extensive set of requirements than member-level participants, since any security breaches on their side could affect broad parts or even the whole IDS ecosystem.

In order to reduce the effort and costs for the certification, the requirements are based on existing schemes and standards for information security (management), namely, IEC 27001 [5] and BSI C5 [6].

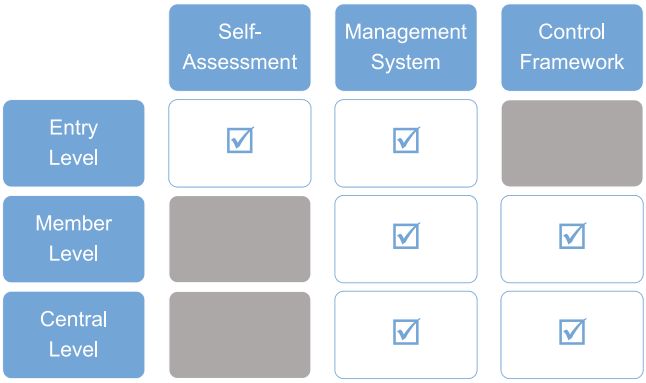

Fig. 9.5 Operational environment certification levels

In addition to security levels, the IDS certification offers three assurance levels that define the depth of the evaluation:

- As a low access barrier, the IDS certification allows the *Self-Assessment* as an assurance level for the *Entry Level*. The applicant fills out and signs a self-assessment concerning the fulfillment of the defined requirements, and, without involving an evaluation facility, the IDS Certification Body reviews the document.
- For a *Management System* evaluation, the evaluation facility assesses whether a company has defined the processes and structures necessary and whether the employees are aware of and following those processes. The evaluators review documentation and process definitions, conduct interviews, and audit the company during on-site visits.
- Companies can utilize control frameworks in their company to ensure and document adherence to the defined management processes. After a certain amount of time after such control mechanisms have been established at the company, their correct application can be verified by a *Control Framework* evaluation. For that purpose, the evaluation facility conducts a *Management System* evaluation as described above and additionally reviews the control mechanisms and evidence for the application of these mechanisms based on random samples.

The described assurance levels and security profiles can be combined in six ways as depicted in Fig. 9.5. Whenever applicants want to get certified, they can decide which security and assurance level they want to address. In reverse, each IDS participant can decide which level they expect from their communication partners as detailed in Sect. 9.5.2.

9.4 Connector Identities and Software Signing

The IDS connect different organizations and components with each other. The trust required for the conducted communication is established by a PKI that manages identities for users and devices and is utilized to implement a technical representation of the certification process described previously in Sect. 9.3.

Figure 9.6 shows the PKI which builds the identity infrastructure necessary for achieving the following purposes:

- Create and manage digital identities for persons, e.g., for employees of an evaluating company.
- Create and manage digital identities for devices.
- Assign ecosystem roles to person identities, e.g., as evaluator of components.

The digital identities of users and devices are bound to the possession of a private key matching the public key contained in their X.509 identity certificate. In the depicted identity ecosystem, the *User Certificate Authority (CA)* issues user

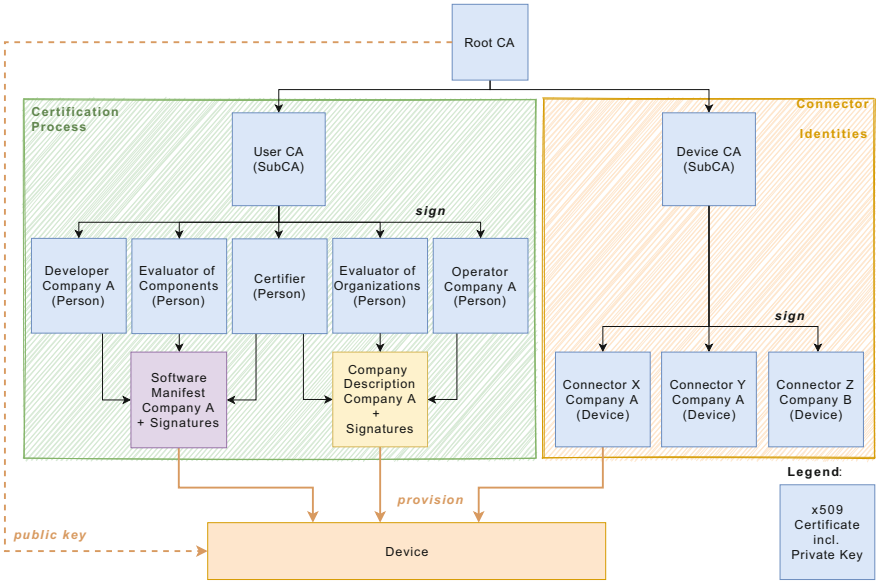

Fig. 9.6 Public Key Infrastructure for building trust in data spaces

identities and assigns ecosystem roles as described in Sect. 9.4.1. The *Device CA* issues device identities which are combined with signed company descriptions as described in Sect. 9.4.2. The signed software manifests necessary for ensuring the connector integrity are detailed in Sect. 9.4.3.

The combination of device identities, signed company descriptions, and software artifacts enables building of trust in the ecosystem. Every device is provisioned with its device certificate, a company description of the operator, the related software manifests for all deployed components, and the public key of the *Root CA* to validate certificates of other party's devices. Optionally, there might be more than one *Root CA* in the ecosystem.

9.4.1 Technical Implementation of the Certification Process

The certification process described in Sect. 9.3 is used to ensure that all components and organizations in the IDS adhere to defined standards. For the verifiability of successful certifications, the IDS include a technical implementation of the certification process that is depicted in the green box in Fig. 9.6. Every person participating in this process receives an identity certificate and owns a private key that allows signing of software manifests and company descriptions. A successful certification process is proven by having these information artifacts signed by multiple parties. As an example, the steps for the component certification are detailed in the following:

- The developer of a software artifact signs the initial software manifest in its role as developer and certification applicant.
- The evaluator of the component signs the software manifest after successful evaluation.
- The certifier signs the software artifact after validating the evaluation report.

The certification for operational environments is conducted in a comparable way to receive a company description with multiple signatures.

The process is designed to create a chain of trust rooted in the unforgeable identity of the applicant. The additional layers of trust are created by an evaluator who verifies that all requirements for the desired trust level are satisfied. Another layer is added by the certifier that ensures the quality and conformity of the evaluation.

9.4.2 Connector Identities and Company Descriptions

Each connector needs to be able to uniquely identify the other connectors in the IDS. For that purpose, each connector instance is provided with a technical identity. This identity is issued by a dedicated *Device (Sub)*CA. Each device is owned by a specific company. The device certificate creates the link to the company description through the company identity. This company description ensures a trustworthy operational environment and is described in the next section. Certified software components can be deployed on multiple devices. Overall trust relies on a combination of device identities, company descriptions, and software artifacts including software signatures.

The company responsible for hosting a connector is described in a company description that has been validated and signed by three independent parties, i.e., the applicant, evaluator, and certifier. It must, at minimum, contain these items:

- The company name.
- The location of the head or branch office.
- The certification level derived from the operational environment certification process.
- The expiry date of the company description.
- An endpoint which may be queried to get a current status of the company description to allow the revocation of the description.

9.4.3 Software Signing and Manifests

For assessing the trustworthiness of the software running on another connector, it is necessary to get information regarding the software stack and the certification it passed. Thus, each software component on a connector needs to be uniquely identifiable using a software manifest that has been validated and signed by all

three parties directly involved in the component's certification process. This manifest contains details to allow an assessment of the components and utilizes cryptographic hashes to uniquely identify the software component belonging to it. In detail, the software manifest must contain at least the following items:

- Cryptographic hash(es) to identify the components which are described by this manifest.
- A unique identifier for the developer of this software artifact.
- An artifact identifier that allows tracking of different versions.
- A version number.
- A classification of the artifact type, e.g., boot loader, kernel, and protocol adapter.
- Functionalities provided by this artifact.
- Usage control policies that can be enforced, e.g., deletion of data after a specified time period.
- The certification level derived from the component certification process.
- The expiry date of the software manifest.
- An endpoint which may be queried to get the current status of the manifest to allow the revocation of the manifest.

Based on the manifests, the software integrity of all component parts becomes verifiable to other connectors and the communication partners are able to assess the risk and possible consequences of sharing their data with this connector.

9.5 Connector System Security

In addition to the certification process as well as the connector identities and software signing, the third building block is the system security of the connectors where data is processed, and usage control must be enforced.

The four core functionalities of a connector are:

1. Providing a *runtime* for isolated data processing apps implementing the IDS information model.
2. *Communication* between connectors with authentication and remote integrity verification.
3. Persistent *storage* of confidential and integrity protected data.
4. *Usage control enforcement.*

These functionalities are implemented in a set of core components depending on the device type (embedded system, gateway or edge device, server, virtual machine, cloud) and the use case with its required security level. Details on possible architectures are provided in Sect. 9.5.1. All components with access to the asset *data* must be trustworthy and their trustworthiness must be remotely verifiable by other connectors before transmitting the data. The utilized communication protocol is defined in Sect. 9.5.2.

9.5.1 Trusted Computing Base

The Trusted Computing Base (TCB) of a connector is the set of all hardware and software components that are critical to the confidentiality and integrity of the transmitted and processed data. Typically, only one or at least a few vulnerabilities suffice to break the system's security. Therefore, a common goal is to reduce the complexity and the size of the TCB which also results in a reduced evaluation effort in the course of the certification process described in Sect. 9.3.

Figure 9.7 shows four common connector system architectures. The system architectures on the top use process isolation mechanisms to protect the data processing apps and IDS functionalities of a connector. The system architectures below use system virtualization with a hypervisor for isolation. The system architectures on the right side make use of an exemplary hardware feature to reduce the TCB. Components in *blue* directly process the data to be protected. Components in *red* are part of the TCB and components in *gray* are not part of the TCB. In the following, we first introduce the components followed by an introduction to the four common connector system architectures.

The components of a connector are as follows:

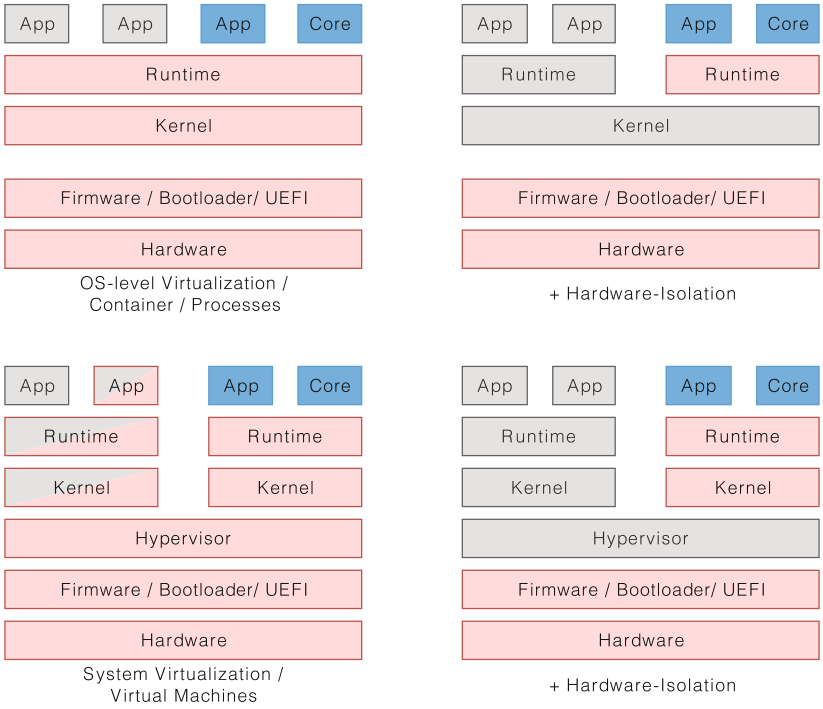

Fig. 9.7 Trusted computing base for IDS Apps and Core

- *Apps* implement all functionalities that are not needed to start other apps. This includes core functionalities required on all connectors and use case-specific features.
- The *Core* is a specific app. It implements the communication interfaces for data exchange with other connectors. This includes an implementation for remote attestation described in Sect. 9.5.2. Encrypted communication between connectors terminates here.
- The *Runtime* implements functionalities to bootstrap apps in a system. It might be small, implementing only a minimal set of functionalities used by the apps. The runtime is a user space component and always part of the TCB.
- The *Kernel* connects user space software to the hardware of a connector. It typically implements memory management, resource management, and device management. User space components use *system calls* to interact with the kernel.
- A *Hypervisor* allows to run multiple *virtual machines* on one physical machine providing a standardized interface to the kernel. In the IDS this may have advantages regarding the deployment, migration, and management of connectors.
- *Firmware/Bootloader/UEFI* includes all software components to bootstrap a system and to initialize hardware before the kernel is started. In interaction with the hardware, these software components bootstrap the trust in the system by implementing a trust anchor and building the root of trust.
- *Hardware* includes all hardware components in the system with access to unencrypted data of apps. This especially includes the processor. It might include a Secure Element (SE), e.g., implementing a Trusted Platform Module (TPM) [7]. It usually does not include hardware devices for persistent storage of data or network interfaces. It might not include Random Access Memory (RAM) if data is always stored encrypted. The hardware must always provide functionalities to bootstrap the trust in the system.

The four connector system architectures presented in Fig. 9.7 are:

- *OS-Level Virtualization/Containers/Processes*: The top-left subfigure shows a typical architecture using OS-level virtualization, i.e., containers. In this case, the *Runtime* includes user space processes running in the root namespace on top of the *Kernel*. The TCB includes *Apps*, *Core*, *Runtime*, *Kernel*, *Firmware/Bootloader/UEFI*, and *Hardware*.
- *OS-Level Virtualization/Containers/Processes + Hardware isolation*: The top-right subfigure shows a possibility to reduce the TCB of the previous architecture using a Trusted Execution Environment (TEE) like Intel Software Guard Extensions (SGX) [8] or ARM TrustZone [9]. RAM used by the TEE is encrypted respectively only accessible by the software components inside the TEE. Functionalities like drivers for persistent storage or network interfaces are kept outside of the TCB. In this case, the TCB includes *Apps*, *Core*, the *Runtime* inside the TEE, *Firmware/Bootloader/UEFI*, and *Hardware*.
- *System Virtualization/Virtual Machines*: The bottom-left subfigure shows a typical architecture using system virtualization, i.e., virtual machines running on top of a hypervisor. In this case, the TCB includes *Apps*, *Core*, *Runtime*, *Kernel*,

Hypervisor, *Firmware/Bootloader/UEFI*, and *Hardware*. Depending on the implementation, the hypervisor might be integrated into a *Kernel* or one virtual machine might be privileged in the system. In such cases, the *Kernel*, *Runtime*, and one or more *Apps* of this privileged virtual machine might have access to the data processed in the virtual machine running the IDS *Apps* and *Core* and therefore might be part of the TCB as well.

- *System Virtualization/Virtual Machines + Hardware isolation*: The bottom-right subfigure shows a possibility to reduce the TCB of the previous architecture using extensions like Intel Trust Domain Extensions (TDX) [10] or AMD Secure Encrypted Virtualization (SEV) [11]. RAM used by the virtual machine is encrypted respectively only accessible by the software components inside the virtual machine. In this case, the TCB includes *Apps*, *Core*, the *Runtime*, *Kernel*, *Firmware/Bootloader/UEFI*, and *Hardware*.

9.5.2 Remote Attestation

In Sect. 9.3 we described the certification and evaluation process of components and operational environments, and in Sect. 9.4 we described how this is mapped to a PKI, manifests, and cryptographic signatures. Afterward we described in Sect. 9.5.1 which components of a connector are critical to its security. The final step is the verification of the identity and trustworthiness of the connector's TCB before transmitting data to it. To achieve this, both communicating connectors perform a remote attestation before any data is exchanged.

Figure 9.8 shows a simplified sequence diagram for the connection establishment up to the first data exchange. The following steps are performed:

- *Hello Message (incl. Nonce)*: In the first step, a hello message including a random number is sent. This number is used to guarantee freshness for the following messages and, thus, prevent replay attacks.
- *Proof of Connector Identity*: In the next step, the connectors identify themselves by using their X.509 certificate and their company description. The private keys of their identity certificates are used to establish the communication channel. Both the signatures of the connector-specific X.509 certificates and the signatures of the company description can be verified up to the *Root* CA. After this step, the identity and the operational environment certification level are known.
- *Proof of Connector Integrity*: In this step, the integrity of both connectors is verified. Each connector implements one *prover* to prove its integrity and multiple *verifiers* to verify the integrity of all sorts of connectors. The prover sends all signed manifests describing its connectors TCB and a prover-specific description of the system state. This typically includes a hash chain beginning with the root of trust for measurement followed by measured hash values representing all components in the secure boot chain. This information is then signed by the prover with its private key in a secure manner. The verifier will then verify the signature,

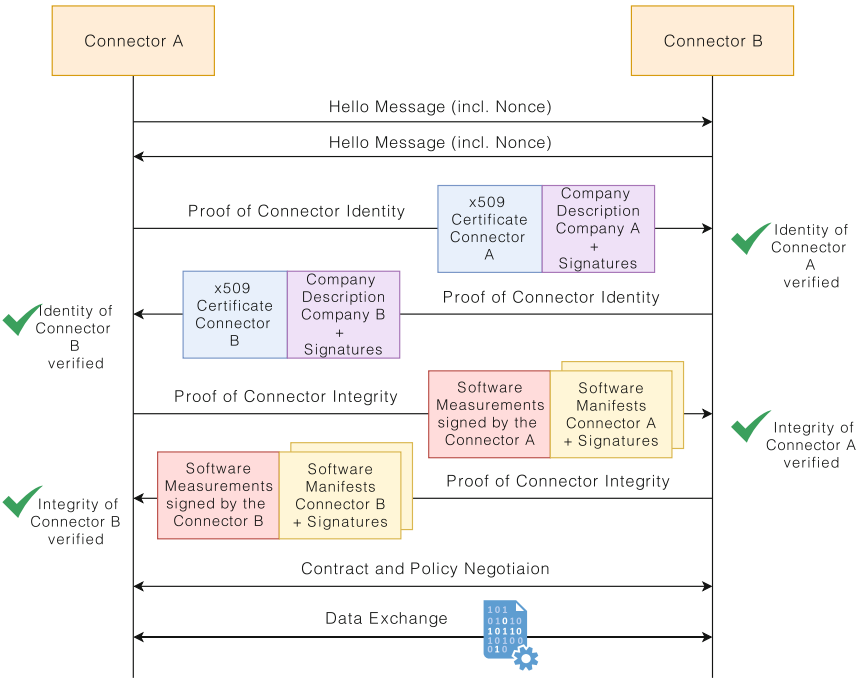

Fig. 9.8 Communication between connectors in the IDS

the freshness, and the hash chain of the software artifacts. Afterward, for each artifact the manifest including the signatures from the developer, evaluation facility, and certification body is verified up to the *Root* CA. After this step, the trustworthiness and the component certification level of the whole TCB are known.

- *Contract and Policy Negotiation*: Based on the established secure and trustworthy communication channel, the two communication partners can negotiate the (legal) terms for data exchange and the required usage control policies.
- *Data Exchange*: In the final step, data can be exchanged.

9.6 Conclusion

In this chapter, we presented the overall security concept for building trust in data spaces enabling data sovereignty and usage control enforcement. We defined data confidentiality and the integrity of the connector stack, which implements the enforcement of usage control policies, as our primary protection goals. To reach these goals, we introduced the three building blocks:

- Certification process.
- Connector identities and software signing.
- Connector system security.

The IDS certification process described in Sect. 9.3 implements a multiple eye principle to gain comparable and trustworthy information concerning the fulfillment of certification requirements for connectors and companies. Both types of certification are technically mapped to a PKI and digital signatures for company descriptions and software manifests in Sect. 9.4. In combination with an X.509 identity certificate for each connector instance, the company description can be used to prove the identity of the connector and its operating company. Additionally, the usage of certified software can be proven to other connectors by using remote attestation utilizing software measurements and signed software manifests. Based on the verification of the identity and integrity of the communication partner, each participant in the IDS is enabled to sovereignly decide with whom and under which conditions they want to share their data. Finally, we have shown multiple connector architectures with small TCB in Sect. 9.5 to reduce costs for the evaluation and certification.

References

1. IDSA. (2019a). *Framework for the IDS certification scheme, Version 2*.
2. DIN. (2020). *DIN SPEC 27070:2020–03 Anforderungen und Referenzarchitektur eines Security Gateways zum Austausch von Indu triedaten und Diensten*.
3. IDSA. (2019b). *IDS reference architecture model version 3.0*.
4. ISO. (2019). *IEC 62443–4-2:2019 Security for industrial automation and control systems - Part 4–2: Technical security requirements for IACS components*.
5. ISO. (2013). *IEC 27001:2013 Information technology — Security techniques — Information security management systems — Requirements*.
6. BSI. (2018). *Cloud Computing Compliance Controls Catalogue (C5)*.
7. TCG. (2014). *Trusted Platform Module (TPM) 2.0*. Library Specification, Family 2.0, Level 00, Revision 01.16.
8. Costan, V., & Devadas, S. (2016). *Intel SGX explained* [Cryptology ePrint Archive, Report 2016/086].
9. Pinto, S., & Santos, N. (2019). Demystifying arm TrustZone: A comprehensive survey. *ACM Computing Surveys., 51*, 1–36.
10. INTEL. (2020). *Intel Trust domain extensions. Technical report.*
11. AMD. (2020). *Secure encrypted virtualization API version 0.24. Technical report.*