Automated Compliance

Gaia-X Institute position paper

07 December 2022

Executive summary

Compliance related to digital sovereignty is a central concern to Gaia-X. The purpose of this paper is to contribute towards a conceptual and terminological framework for developing automated compliance in the context of Gaia-X. Compliance here refers to conformance of a system to a set of rules and regulations, or to conformance with agreements among parties. Automated compliance, as understood here, refers to technologies (algorithms, software, hardware) which can assist either in achieving compliance in a system, in checking compliance of a system, or in enforcing compliance of a system. Compliance automation is subject to inherent limitations, both for technological reasons and for reasons of jurisprudence and legal fundamental principles. Still, raising the level of compliance automation as far as possible is an essential tool to reach monumental goals of Gaia-X, for reasons of efficiency, scalability, and reliability. Compliance automation technology should provide information and artefacts (for example: facts, logs, proof, certificates, evidence) of legal relevance. In addition to the development of compliance automation technology, contributing towards better understanding of the interface between technological and legal notions of compliance is a central area of concern to automated compliance. There is a need for increased R&D devoted to the area of compliance and its automation, both in order to raise the level of automation and in order to understand possible gaps between, on one hand, legal and regulatory systems, on the other hand, means of achieving and enforcing compliance. The notion of Labels provides an essential instrument for extending compliance from a standard core.

Compliance

The purpose of this paper is to contribute towards a conceptual and terminological framework for developing automated compliance in the context of Gaia-X.

Compliance, as understood in this document, refers either (in a narrower sense) to *regulatory compliance*, that is, conforming to a rule, such as a specification, policy, standard or law[1], or (in a more general sense) to compliance with agreements between parties, for example, service level agreements between stakeholders in the market. Of special concern to Gaia-X is compliance with respect to regulations and agreements related to *digital sovereignty*[2], which is at the core of the mission of Gaia-X in the context of the European Data Strategy[3].

Automated compliance as understood here refers to technologies (algorithms, software, hardware) which can assist either in *achieving* compliance in a system, in *checking* compliance of a system, or in *enforcing* compliance of a system. Automated compliance may be regarded as a form of *regtech*, regulatory technology[4], directed at data sovereignty and the areas of concern to Gaia-X.

Compliance automation involves dealing with some of the most challenging problems in computation, and understanding the design space of compliance automation is complicated, requiring a systematic and scientifically informed approach. As will be explained in more detail in this paper, there are limitations to what can be automated in this area, both for inherent (ultimately, mathematical) reasons and for reasons of law. On one hand, it is a consequence of basic results of computer science that not all properties of programs or systems can be automatically verified. On the other hand, no level of automation of compliance can replace jurisprudence or the human factor essentially involved in the legal dimension of compliance. Still, developing compliance automation *as far as possible* is an essential tool for implementing compliance in practice.

---

[1] See Frison-Roche (ed.): *Compliance Tools*. Journal of Regulation and Compliance. Bruylant 2021. 978-2-8027-7040-4 (ISBN). See also, e.g., https://en.wikipedia.org/wiki/Regulatory_compliance

[2] EPRS Briefing on *Digital sovereignty for Europe*, European Union 2020. https://www.europarl.europa.eu/RegData/etudes/BRIE/2020/651992/EPRS_BRI(2020)651992_EN.pdf

[3] European Commission*: COMMUNICATION FROM THE COMMISSION TO THE EUROPEAN PARLIAMENT, THE COUNCIL, THE EUROPEAN ECONOMIC AND SOCIAL COMMITTEE AND THE COMMITTEE OF THE REGIONS. A European strategy for data.* Brussels 19.2.2020. https://ec.europa.eu/info/sites/default/files/communication-european-strategy-data-19feb2020_en.pdf

[4] See e.g., Wikipedia: https://de.wikipedia.org/wiki/Regulatorische_Technologie, and World Economic Forum white paper on *Regulatory Technology for the 21st Century*, March 2022, https://www.weforum.org/whitepapers/regulatory-technology-for-the-21st-century/

The purpose of this paper is to contribute towards a conceptual and terminological framework for developing automated compliance in the context of Gaia-X.

The need for compliance automation

Regulation with respect to digital sovereignty, including the EU Data Governance Act[5] and the EU Data Act[6], is increasing at rapid pace in response to societal concerns that are central to European values and to Gaia-X. Increasing levels of regulation lead to the need for corresponding procedures for *implementing* (achieving, checking, enforcing) regulatory compliance. In order to realistically implement compliance, it is increasingly necessary to develop tools to automate compliance implementation, for the benefit of all stakeholders.

Legal and technical notions of compliance

In addition to inherent technical challenges for automating compliance, a further fundamental challenge arises from the necessity to consistently understand the notion of compliance both from a legal perspective and from a technical perspective: Automated compliance is a tool to help achieve, check, or enforce compliance properties of *technical* systems. These properties ultimately are defined by or follow from *legal* and regulatory systems. Already at the terminological level, it can be challenging to talk about both aspects at the same time without risk of misunderstanding. For example, the term “procedure” means something different in law and in computer science (although the meanings might be related, which may only increase risk of misunderstanding). This document is written from a mostly technical (computer science) perspective. Further work is necessary to clarify the interface between legal and technical notions of compliance.

Compliance by design, ex-ante, ex-post

Compliance by design refers to *modes of construction* of systems or components towards achieving compliance. For example, a smart metering system may use only sensor technology that has been approved *a priori* for the purpose. Or, a software system may use cryptographic components that have been certified for certain security and privacy levels. The distinction between *ex-ante* and *ex-post* refers to different *modes of regulation*[7]. For

[5] https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52020PC0767
[6] https://ec.europa.eu/commission/presscorner/detail/en/ip_22_1113
[7] See Frison-Roche (ed.): *Compliance Tools*. Journal of Regulation and Compliance. Bruylant 2021. 978-2-8027-7040-4 (ISBN).

example, in the area of regulation of digital markets[8], ex-ante regulation may refer to policies adopted to prevent anti-competitive behaviour, whereas ex-post may refer to policies for enforcement or punitive action once anti-competitive behaviour has occurred. In the context of compliance, broadly speaking, *ex-ante compliance* stipulates policy and behaviour necessary to achieve compliance, whereas *ex-post compliance* refers to regulations dealing with cases of non-compliance.

An example: smart metering

A smart meter[9] is an electronic device that records information about consumption of resources (for example, gas, water, or electric energy). For example, the meter could record voltage levels, current, and power factor, and it could record dates and time intervals of such measurements, which typically happen in near real-time. Smart meters may communicate information to consumers (e.g., for understanding consumption patterns), and to suppliers (e.g., for system monitoring and customer billing). Smart meters enable two-way communication between the consumer and the supplier, in an automated manner provided by the smart meter.

The deployment of smart metering systems is currently receiving renewed attention due to large-scale societal factors and goals, including transition of energy away from fossil resources and towards reduction of CO2 emissions, digitalisation towards intelligent energy systems, reducing resource (energy) consumption to cope with crises of shortage, reducing economic or political dependence on certain countries exporting energy resources.

From a computing perspective, smart metering systems are data intensive, distributed (and in part cloud-based) multi-stakeholder systems, which are subject to regulation. Smart metering systems illustrate the need to regulate complex, data intensive systems in order to mediate possibly conflicting interests among stakeholders, for example privacy concerns of consumers, (cost-) efficiency of providers, policy goals of public bodies. Thus, in 2012 data protection issues were the subject of regulatory concern within the European Commission in preparation for the roll-out of smart metering systems[10], and smart metering systems have been subjected to specific regulatory rules, for example[11]:

- Without explicit approval by the consumer, all data-gathering and use is restricted to the bare minimum required for the energy system to work

[8] See for example OECD (2021), *Ex ante regulation of digital markets, OECD Competition Committee Discussion Paper.* https://www.oecd.org/daf/competition/ex-ante-regulation-and-competition-in-digital-markets.htm

[9] https://en.wikipedia.org/wiki/Smart_meter, https://de.wikipedia.org/wiki/Intelligenter_Z%C3%A4hler, https://fr.wikipedia.org/wiki/Compteur_communicant

[10] Opinion of the European Data Protection Supervisor on the Commission Recommendation on preparations for the roll-out of smart metering systems, 2012. See also: https://edps.europa.eu/data-protection/our-work/publications/techdispatch/techdispatch-2-smart-meters-smart-homes_en.

[11] This example is from The German Ministry of Economic Affairs and Energy*: Federal The German Smart Metering – Subject to Stringent Data Protection and Security Rules*.

- The intervals at which the meter is read have been designed to be long enough to prevent any conclusions being drawn about user habits
- No data will be transmitted unless it has been anonymised, pseudonymised, or aggregated
- Data will be processed *in situ*, right on the consumer's premises
- Energy data will be passed on to as few parties as possible
- It will be mandatory for data to be deleted within specified time periods
- Consumers will be able to monitor and verify all communications and processing steps at all time
- It will be easy for consumers to enforce their right to object and to data being deleted or corrected
- Consumers will still be able to choose the tariff that suits them best

There are many concrete scenarios of interest to citizens that can be brought about by smart metering, for example:

- If you claim a reduction of your invoice because you have not used your washing machine during peak hours, only a smart meter measuring your electricity consumption every half an hour can confirm you are compliant and bring a pre-constituted legal proof of that.
- If you are prepared to be part of electrical erasure you can voluntarily reduce your subscribed power for instance from 9 kva to 3 kva through the smart meter to get a discount, and then you will have to arbitrate between airconditioning and electronic vehicle charging, otherwise your circuit breaker will cut you off.

Notice that the example not only illustrates the need for compliance in order to impose *limitations* on the use of data obtained by smart metering systems. It may also be used to illustrate the need for compliance in order to *enable* the use of such data in ways which may be deemed desirable. For example, one might consider using smart metering for incentivising responsible ecological behaviour of citizens, by enabling discounts to customers who contribute to reducing energy consumption. Compliance of the smart metering system according to regulation of such a scenario would be needed, both as a matter of policy and from the standpoint of civic acceptance.

Compliance and automation

Distinction between semantic and procedural notions of compliance properties of systems. Inherent limits of automation. Automated compliance as a "monumental goal". The interface between technological and legal notions of compliance.

Semantic versus procedural notions of compliance and limits of automation

In the context of computational (in particular, software-based) systems, it can be useful to distinguish between two distinct, but related, notions of compliance properties. In one sense, a compliance property may be understood as a *semantic* property of a system or program[12]. In computer science, a semantic property pertains to the *behaviour* of the computational system. An example of specifying such a property of a program might be: "This program computes the square root function on natural numbers". This specification says that the program, when given a natural number *n* as input, will produce the square root of *n* as output. In this case, the specification refers to *all possible* input-output behaviours of the system (for *all* numbers *n*, the output will be the square root of *n*). Notice that there are infinitely many such behaviours, which are comprised by the specification. Semantic compliance can be understood as the "ground behavioural truth" for compliance, referring to all possible behaviours of the system. Semantic compliance properties may be specified via a set of *semantic rules*, compliance meaning behavioural consistency with the rules. For example, the rule for the smart metering system

- "No data will be transmitted unless it has been anonymised, pseudonymised, or aggregated"

is a semantic rule requiring *all* behaviours of the system to have the property that they do not transmit data unless it has been anonymised, pseudonymised, or aggregated.

Semantic properties of systems may be complicated to ascertain, because they may refer to infinitely many behaviours of the system, and it may not be possible to check such properties exhaustively by testing the system (meaning, executing the system on finitely many test cases). In general, semantic properties of programs are *algorithmically undecidable*[13]. Undecidability of a program property means that there cannot (for mathematical reasons) exist an algorithm which always correctly determines whether a

[12] The word "program" here is used to mean a piece of software, that is, a piece of text written in a programming language which can be executed on a computer. The word "system" as used here typically refers to computational entities composed of many hardware- and software components.
[13] See, e.g., the classical text by Martin Davis: *Computability and Unsolvability*. McGraw-Hill 1958. More specifically for the present context, the relevant result of theoretical computer science is Rice's Theorem which says, essentially, that any non-trivial extensional property of programs is undecidable. A property is trivial if it is the empty set or the universal set. A property is extensional, if it only depends on the input-output behaviour of the program. See e.g. https://en.wikipedia.org/wiki/Rice%27s_theorem.

given program has the property. For example, the halting property of programs is famously undecidable[14]: Given any program, to decide whether it ever halts (stops) or not. Some (even some quite simple-to-state) properties of programs therefore cannot be automatically checked with complete precision in total generality, such as for example to determine, given any program, whether it could ever attempt to perform a division by 0. It is a consequence of these basic results of computer science that:

- *checking arbitrary semantic compliance properties of systems cannot be fully automated in general*

Hence, when we talk about "automation of compliance", "automated compliance" etc. it must always be understood that automation may be only partial or may only pertain to certain restricted aspects of the system. That being noted, such partial automation may still be extremely useful and economically attractive.

In another sense, compliance may be understood as a *procedural*[15] property referring to adherence to a set of *procedural rules* regarding various aspects of a system. Such properties may state that a system is constructed or used according to certain procedures (e.g. procedures for assembly, programming, deployment, or operating and maintenance), or they may characterise the originator of the system (e.g. authenticating the origin of a part of a system), or they may characterise ways in which a system has been inspected or certified (e.g. following certain audit procedures). In the smart metering example, the rules

- "Consumers will be able to monitor and verify all communications and processing steps at all time"
- "It will be easy for consumers to enforce their right to object and to data being deleted or corrected"

can be understood in a procedural way (the smart metering system offers appropriate monitoring services to consumers, and there are procedures in place for consumers to make certain claims).
Certification of procedural compliance may often be considered a matter of making sure that specified design guidelines, engineering guidelines, auditory procedures, or other regulatory rules have been duly followed by the relevant parties (e.g., producers of the system, system vendors, users of the system, etc.).

In most contexts, semantic notions appear mixed with procedural notions. Typically, rules tend to be expressed with reference to some semantic properties and some procedural rules. The intended meaning of such expressions may for example be that certain procedures should be applied in order to ascertain (with some level of confidence) that the semantic rules are fulfilled. In the smart metering example, the rule

[14] This is Alan Turing's famous result from 1936.
[15] The term "procedural" is not to be understood in a legal sense here but refers rather to what is known as "engineering procedures".

- "Without explicit approval by the consumer, all data-gathering and use is restricted to the bare minimum required for the energy system to work"

can be understood as a mixture of procedural and semantic notions: The approval of the consumer is a procedural idea (e.g., the consumer has filled in and signed a certain form), and the reference to data-gathering is a semantic notion referring to system behaviour. In general, a basic problem faced in certification of compliance can be understood as one of *procedural approximation of semantic truth*:

- To define a set of procedural rules that ensure with some reasonable level of confidence that a set of semantic rules are likely to be fulfilled.

Even if checking compliance with a set of procedural rules may be theoretically possible, perfect compliance checking could still be a *practically unattainable* goal, depending on the scenario. For example, in some scenarios, complete certification of compliance could require inspection of the entire technology stack, from the software application all the way down through the systems level and through the levels of hardware. In the smart metering example, we have a scenario spanning many layers including user-level software (websites, apps etc.), communication hardware and software, service- and provider-side server software, sensor software and hardware. At any level in this network of subsystems one could theoretically imagine sources of violations of regulation for any number of reasons (malice, inattention, incompetence, software bugs, etc.).

Automated compliance

The goal of compliance certification as understood and pursued here is to enable reasonable levels of trust at reasonable levels of cost obtaining certification. A reasonable level of trust may be one that ensures that trust violations have a high probability of incurring high cost (either in terms of operationalisation or in terms of penalty) on violators. Achieving reasonable levels of *automated* compliance may be considered a tool towards realising a "monumental goal"[16].

Automated compliance may refer to automation of different aspects of compliance certification, including

- *Construction*. Certifying the application of compliance-by-design rules.
- *Verification*. Certifying that compliance of a system is verified, validated, tested.
- *Procedures*. Certifying that compliant engineering and operating procedures are in place (effective procedures are defined and in use).

[16] See Frison-Roche, Gouriet, Tardieu: *Compliance and consequences on the Gaia-X labeling framework*.

Motivations and goals for automated compliance include:

- Trust
- Scalability and efficiency
- Cost reduction

These goals may be mutually conflicting. For example, implementing a monitoring system to achieve automatic compliance checks at run-time may be costly. A central objective is:

- To analyse and structure the design space of automated compliance with the objective of identifying points in the space that may ensure reasonable levels of trust at reasonable cost.

Trust levels may vary, and cost levels to obtain certification may vary accordingly. This is part of the rationale behind Gaia-X Labels.

Judging from a broad orientation in state-of-the-art methods of automation (see Taxonomy below), it is to be expected that automation of compliance will have to be composed from a mix of *technological components* including:

- Compliance by *design*
- Compliance by *testing*
- Compliance by *monitoring*

As with security, an important aspect of implementing compliance in practice is "*compliance culture"* referring to, broadly speaking, the human factor (human behaviour and culture), defining values, competencies, training, cultivating awareness, near miss reporting, breach reporting, continuous improvement, etc. Although we do not here understand this aspect directly as a technological component *per se*, it must be regarded as a possibly necessary component in *implementing* compliance in practice. Certain forms of automation may require certain aspects of culture in order to be effective. An example (from security) is the use of passwords and authentication technologies which may be rendered ineffective in the absence of a suitable culture (e.g., of creating strong passwords). In addition, the cultural aspect may be a target for automation in providing *support* for human-based processes (for example, partially automating a process of reporting).

Compliance can be assessed in various phases of the life cycle of a system: from conception to design, engineering and deployment, operation and maintenance. These phases split into two major parts: before the actual use (ex-ante: before the fact) and during operation (ex-post: after the fact). *Ex-ante compliance* is an important legal concept[17], the technological counterpart of which may be understood as "compliance by design and by

[17] See Frison-Roche: *Compliance Tools*. Bruylant 2021. 978-2-8027-7040-4 (ISBN).

testing". This concept is central to any regulatory compliance system[18] that validates a new solution before "entry in the market". And in most cases the regulator imposes an ongoing surveillance, or monitoring, when solutions are sold and in actual use, requiring mechanisms for reporting or alerting, corrective and preventative actions, or continuous improvement. Procedures of *ex-ante* regulation and compliance are typically contrasted with *ex-post* procedures[19].

Currently, Gaia-X compliance is centred around architectural concepts, self-descriptions, the extension of verifiable credentials via linked data, and the concept of labels. Naturally, there are still some open issues within the current scope of Gaia-X Compliance and the Gaia-X Trust Framework, some of which are important for automation. Some technical issues will be associated with legal issues[20].

Some central questions for furthering automated compliance in the Gaia-X architecture include:

- What are main open technical and legal issues in the current design and within the current scope of Gaia-X Compliance and the Gaia-X Trust Framework?
- Which currently defined areas of Gaia-X Compliance should be prioritised for automation? Which are the exact computational problems underlying those areas?
- How to automate the extension of compliance properties "upwards" into the software layers so that participants (including in particular, application or service developers) can obtain compliance certification at reasonable cost.

Compliance automation and legal notions of compliance

There are at least three broad dimensions of compliance: socio-political, legal, and technological. We consider here some general points pertaining to the interface between compliance technology and the legal dimension, in particular with a view to the topic of automation of compliance.

It is important to be aware that no level of automation of compliance can *replace* jurisprudence or the human factor essentially involved in the legal dimension of compliance, which is a new branch in legal systems[21]. It has been pointed out above that, already for purely mathematical reasons of computability, compliance properties cannot in general be *fully* automated. From a *legal* perspective, a similar conclusion follows, but for different reasons. Compliance cannot be fully automated, because *jurisprudence* and *the*

[18] E.g. OECD (2021*), Ex ante regulation of digital markets, OECD Competition Committee Discussion Paper*. https://www.oecd.org/daf/competition/ex-ante-regulation-andcompetition-in-digital-markets.htm
[19] https://mafr.fr/en/article/ex-ante-ex-post2/
[20] An example provided by P. Gronlier: It is foreseen that trust extension can happen automatically by extending a key chain or signing a new key pair with an existing eIDAS key. Even if the original eIDAS-signature is legally binding, it may be an open question whether the machine-generated key pairs and signatures are legally binding, as of the current legal situation.
[21] See Frison-Roche: *Compliance Tools*. Bruylant 2021. 978-2-8027-7040-4 (ISBN).

legal system cannot be fully automated. One cannot under current jurisprudence imagine judges being substituted by algorithms: Judges constitute, *by law*, an essential human factor in the legal system. Apart from the fact that this state of affairs is grounded in law itself, it is also understandable from considerations of the limitations of technology. Thus, for example, it is an essential task of a judge to apply the law according to its "spirit". We do not have (and perhaps will never have) access to technology which would enable automation of that kind of reasoning[22].

Even if we restrict attention to specific compliance properties which could, in principle, be completely automated (for example, verifying that the data flow between a smart metering system in a home and a server is properly encrypted), that still would not completely eliminate the human and political aspects of compliance from the legal perspective. For example, it could always happen that compliance of a system is contested by a stakeholder. This could even happen by the compliance check itself being contested (the verification is disputed for being erroneous or incomplete). Such cases could end up in court and hence before a human judge. The situation could also be reversed by a political or regulatory body in a different policy spirit, which algorithms cannot catch.

In view of the foregoing considerations, the following appears to be a useful general formulation of the goals of automated compliance vis-à-vis its legal implications:

- Automated compliance procedures and algorithms should produce *evidence* (e.g., traces, logs, certificates, facts, proofs, etc.) of *legal relevance*, including such evidence that is necessary before regulatory and supervisory bodies and courts because the burden of proof is on the stakeholders in the market.

Correct understanding of this statement includes a number of aspects, which are supported by the foregoing analysis:

- *Evidence* includes a range of formal artefacts depending on the case at hand. For example, facts can be produced by archiving measurements, results, documents or any type of digital transactions, traces (run-time logs) can be produced by monitoring procedures, certificates could be test results (possible aggregated and abstracted) produced by certified testing procedures, proofs could be handmade proofs of compliance of algorithms of limited scope and/or their implementations (for example: correctness of an encryption algorithm and/or its implementation), or proofs could be machine-generated formal proofs that can be formally checked.
- *Legal relevance* is open to interpretation and can mean different things depending on the case at hand. It should be seen as part of the effort towards automated compliance to clarify, to the extent possible, legal implications of the evidence produced by automated compliance procedures.

[22] One can have philosophical discussions about whether future AI-technologies might reach the level of human common sense reasoning, but we forego such discussions here.

The consideration of the legal dimension of compliance raises questions, including *liability questions pertaining to automated compliance tools*. If Gaia-X wants to offer automated compliance tools to stakeholders, the legal implications of doing so (including questions of contracting, agreements, and liability) will need careful scrutiny.

The Gaia-X Trust Framework

Summary of the main technical concepts of the current state of the Gaia-X Architecture entering into the Gaia-X Trust Framework, which are relevant for understanding potential automation within the current scope of the framework.

In order to relate in more detail to the technical work in Gaia-X, we briefly summarise the main technical concepts of the current state of the Gaia-X Architecture[23] entering into the Gaia-X Trust Framework[24] which are relevant for compliance automation. These concepts and frameworks are subject to change, and the following should be understood as a snapshot.

The Gaia-X Trust Framework uses *verifiable credentials* and linked data to build a FAIR knowledge graph of verifiable claims from which additional trust and composability indexes can be automatically computed[25].

The Gaia-X Trust Framework builds on Gaia-X Self-Description files following the W3C Verifiable Credentials Data Model, for describing entities in all relevant participant *roles* of the Gaia-X Architecture[26], including Consumer, Provider, Federator, Resource, Service Offering. Gaia-X Self Descriptions may be endowed with a taxonomy and an inheritance structure[27]. Relations between Gaia-X Self Descriptions may be specified by RDF-triples thereby giving rise to a Self-Description Graph[28]. This graph may be extended by so-called edge properties endowing the edges of the Self-Description Graph with additional attributes besides their type such as the origin of the claim, the issuer, and others[29].

The Gaia-X Trust Framework works with four types of *rules* pertaining to: serialisation format and syntax, cryptographic signature validation and validation of the keypair associated identity, attribute value consistency, and attribute veracity verification. The Gaia-X Trust Framework is defined[30] as the process of going through and validating the set of *automatically enforceable rules* to achieve the minimum level of Self-Description compatibility in terms of:

- syntactic correctness
- schema validity

[23] Gaia-X Architecture Document 22.04 Release.
[24] Gaia-X Trust Framework 22.04 Release.
[25] Gaia-X Trust Framework 22.04 Release, p. 3.
[26] Gaia-X Architecture Document 22.04 Release.
[27] Gaia-X Architecture Document 22.04 Release, 4.2.
[28] Gaia-X Architecture Document 22.04 Release, 4.4.
[29] Gaia-X Architecture Document 22.04 Release, 5.4. 1..
[30] Gaia-X Architecture Document 22.04 Release, 6.3.

- cryptographic signature validation
- attribute value consistency
- attribute value verification

Whenever possible, the verification of Self-Descriptions' attribute values is done either by using publicly available open data, and performing tests or using data from Trusted Data Sources. This verification is captured using Verifiable Credentials issued by either of the following Trust Anchors:

- the Gaia-X association when performing live tests
- the owner of the Trusted Data source

Furthermore, it is expected that checking the validity of Self-Descriptions using open data and test data will introduce costs.

Trust anchors are Gaia-X endorsed entities responsible for managing certificates to sign *claims*, which are assertions appearing in Self-Descriptions[31]. To be compliant with the Gaia-X Trust Framework, all keypairs used to sign claims must have at least one of the endorsed Trust Anchors in their certificate chain. At any point in time, the list of valid Trust Anchors is stored in the Gaia-X Registry. Gaia-X builds on eiDAS for electronic identification, authentication and trust services. The Gaia-X Association defines[32]:

- the sets of rules to define the Trust Anchors:
  - Trust Service Providers.
  - Gaia-X Label Issuers
  - Trusted data source for Gaia-X Compliance
- the format of the Self-Descriptions and their compliance rules
- the Gaia-X Labels rulebook.

Currently, in the Gaia-X Architecture Document, *Gaia-X verification* refers to validating signed claims using the Gaia-X Trust Framework[33].

Gaia-X Labels[34] is the Gaia-X concept for optionally extending compliance beyond the standard core level of *Gaia-X Compliance*. Technically, a Gaia-X label is a W3C Verifiable Credential. A *Gaia-X Label* is a key component of the Gaia-X Trust Framework, which has as its stated goal:

[31] Gaia-X Architecture Document 22.04 Release, 4.1. p. 22.
[32] Gaia-X Architecture Document 22.04 Release, 5.2. p. 31.
[33] Gaia-X Architecture Document 22.04 Release, 4.6.2 p. 28.
[34] Gaia-X Labelling Framework

- the development of “a Compliance- and Labelling-technological framework automating all the tests and verifications needed to give a service a specific Label”[35].

The relation between Gaia-X Labels and Gaia-X Compliance is clarified as follows[36]:

- *Gaia-X Compliance* is defined as “the process of going through and validating the set of automatically enforceable rules to achieve the minimum level of Self-Description compatibility in terms of file format and syntax, cryptographic signature validation, attribute value consistency and attribute value verification” (Technical Architecture Document - TAD, 21.09). In that sense, Gaia-X Compliance ensures that the required level of information for users to make decisions is available, and that such information is verified or verifiable. Gaia-X Compliance specifies conditions for a Provider, as well as for the Service Offerings proposed by such a Provider.
- *Gaia-X Labels* “ensure that a predefined set of policy and technology requirements are met (PRD, 21.04). From a technical perspective, Labels are the result of the combination of verified “Self-Description compliant attributes, that individually would be insufficient to support business or regulatory decisions.” (TAD, 21.09).”

Gaia-X Labels are currently organised in 3 progressive *levels*, defined by a set of *compliance criteria*. The criteria that define the different levels are defined in detail in the Gaia-X Labelling Criteria Catalogue[37].

Gaia-X Labels provide a means of abstraction and aggregation for compliance credentials. Using Labels, compliance credentials can be automatically found, linked, aggregated, and transitively extended. Because Gaia-X Labels hide possibly complex compliance properties behind the labels, the concept of labels potentially supports essential technical opportunities for *modularisation* and *separation of concerns* for compliance automation.

In addition to Gaia-X Compliance and the Gaia-X Trust Framework the *Gaia-X Policy Rules* Document[38] contains policy rules, which define “high level objectives safeguarding the added value and principles of the Gaia-X ecosystem. To allow for validation, the high-level objectives are underpinned by the actual requirements of the suitable criteria catalogues, as further specified in the Gaia-X Label and Trust Framework documents.”

[35] Gaia-X Trust Framework 22.04 Release, p. 2.
[36] Gaia-X Trust Framework 22.04 Release, p. 2.
[37] Gaia-X Trust Framework 22.04 Release, p. 4.
[38] Gaia-X Policy Rules Document PRD 22.04.

Taxonomy of automation methods

The following reasoned taxonomy describes some major, currently accessible methods which involve, or may be developed to involve, a significant degree of automation of relevance to Gaia-X compliance. It is not intended to be exhaustive but may be taken as a starting point for future work towards understanding automated compliance technologies in relation to the goals of Gaia-X.

In developing technology for compliance automation for Gaia-X it is important to structure the technical design space. Different technical approaches need to be accompanied by reasoned assessments pertaining to their pro's and con's and their relevance to Gaia-X notions of compliance. The degree to which automation of compliance is currently possible may depend significantly on which aspects of systems and which technical approaches to automation are considered (see text accompanying each item below). The following reasoned taxonomy of automation methods is based on technological aspects of approaches or systems, which are generally not mutually exclusive. For example, most software-based methods currently have elements of testing, or monitoring, or both. The following taxonomy is not intended to be exhaustive but may be taken as a starting point for future work towards understanding automated compliance technologies in relation to the goals of Gaia-X.

Linked data

Linked data is probably the most important structure for *trust chaining* as of the current state of design in Gaia-X. The main problem solved by linked data is to provide the technological basis for creating a graph of transitive, verifiable claims, thereby enabling the computation of chains of trusted credentials extended from trusted sources and their self-descriptions (ultimately, in Gaia-X terms, Trust Anchors). The linked data approach provides mainly procedural certification. The linked structure as such does not itself provide any semantic compliance guarantees, providing a structure for extending trust from trusted sources. The semantic significance of the linked structure depends on the semantic conditions for obtaining credentials, which may, for example, involve tests or monitoring.

Architecture-based methods

The architecture of a system defines the overall design and broad structure of the system and determines many aspects (e.g., stakeholders, types of components, communication infrastructure and topology, data flow, protocols, etc.). Architectural concepts must therefore be at the basis of any operational compliance system and are necessary instruments for achieving compliance by design or ex-ante compliance. Gaia-X Compliance as well as the Gaia-X Trust Framework are grounded in and structured by the Gaia-X Architecture.

Trusted components and trust extension

An interesting area of research and development concerns the idea of automatically extending compliance properties from trusted components into the software application layer (e.g. apps, services). Methods for operationalising such trust extension could open the door to provisioning of trusted component repositories for application developers. A key question to be addressed is:

- How to certify that the way trusted components are used in a software application provides ground for trust extension to the application (or parts thereof).

This question may be addressed with techniques based on testing, monitoring, compilation, languages (DSLs) etc. It is possible that component structure may help automation. For example, may be useful in assembling systems from trusted components according to architectural patterns, such that, for example, certifying monitors and tests become available automatically.

Testing

Together with monitoring-based methods (see below) test-based methods are among the most important currently deployable techniques for partially checking semantic properties of software systems. Test-based methods are necessarily semantically incomplete, since they can only cover finitely many behaviours at any given time. Many (if not most) algorithms and programs are specified to (or supposed to) work correctly over *all* of infinitely many possible inputs. Example: an algorithm to compute the square root of natural numbers must work correctly on all (infinitely many) numbers. But any test can only run a program on finitely many inputs in finite time. Testing can therefore in general only *falsify* correctness properties with certainty: If a program is *incorrect*, then this must manifest itself on *some* input, and a test executing the program on that input can reveal beyond reasonable doubt that this is so. In contrast, *verifying* a correctness property may require infinitely many tests and may hence not be testable in finite time. Still, testing is of paramount importance in practice. Testing is of central importance for extending trust and compliance into the software layer. Testing can be very costly in terms of engineering effort. Developing effective test strategies (with reasonable coverage of relevant properties) can be costly.

An interesting idea for research and development in the context of Gaia-X Compliance is

- To design certified test repositories for various architectural components (e.g. data connectors) together with automated means of test deployment, thereby lowering the cost of testing while heightening the level of trust.
- To design templates and tools that simplify the assessment of test coverage of a component, an application, or an entire system (e.g. percentage of code covered by test cases, percentage of identified behaviours covered, etc).

Monitoring

In addition to test-based methods monitoring-based methods are among the most important currently deployable techniques for partially checking semantic properties of software systems. Like test-base methods, monitoring-based methods are necessarily semantically incomplete, since they can only cover finitely many finite execution traces of a system at any given time. Therefore, monitoring may only provide partial coverage for compliance requirements. Nevertheless, using a risk-based approach, monitoring specifications are typically implemented for the highest risk aspects of a solution. Some properties of programs and systems can be monitored at run-time (see below), and run-time monitoring of a system can ensure that no actual execution of the system violates such properties. Emerging technologies of interest in the area of monitoring include the use of machine learning techniques, for example, to help identify anomalies of system behaviour.

Monitors can often be related to formal specifications (for example, regular expressions) of classes of properties (for example, so-called safety properties) and can in some cases be derived automatically from them (for example, finite state machines derived from the specification of a safety property). *Run-time verification* refers to verification of execution traces using monitors. Challenges for monitoring-based methods include the fact that monitors may change the system under observation, since monitors must typically be implemented by instrumenting the system under observation with additional code. A monitoring system needs to be secured to avoid tampering, as malicious parties might change monitoring logic to filter out signals that are undesirable to them, or to change alert thresholds, or falsify the monitoring information all together. Also, monitors may incur runtime overhead incurring performance degradation of the system. Finally, instrumenting a system with monitors may be costly. If the system changes, the monitoring system may have to change accordingly. Continuous service certification can therefore be challenging[39].

Compliance-as-code

Compliance-as-code does not purport to reduce all aspects of compliance to code (which, as we have seen, would be claiming to do the impossible). Rather, compliance-as-code

[39] See e.g. Greulich, Lins and Sunyaev: *From data to Insights: Leveraging Monitoring Data for Achieving Continuous Certification of Cloud Services*. Twenty-Seventh European Conference on Information Systems (ECIS2019), Stockholm-Uppsala, Sweden.

refers to a notable recent and emerging movement in the software systems engineering field, following onto the various “X-as-code” or even “everything-as-code” movements (such as infrastructure-as-code, data-as-code), which is often positioned as a natural further development of DevOps-approaches. Several companies are offering various solutions marketed under the heading.

The basic idea, in terms of currently available technology, is to provide systems tools for representing and operationalising compliance rules as tests or monitors or both, and possibly using information obtained from them to generate audit reports.

Compliance-as-code as it is currently realised can therefore be seen as a way of using test-based methods and monitoring-based methods to translate compliance and policy rules into software-based automated compliance checks, and therefore pros and cons of those methods can be expected to be inherited. The main innovation contributed so far by compliance-as-code approaches appears to lie in automation towards closing the gap between compliance rule systems and available methods of testing or monitoring.

Language-based methods

Semantic guarantees on the behaviour of software systems can be obtained by the employment of programming languages (general purpose languages or DSLs[40]) whose expressions are restricted to obey given rules which are enforced by the compiler. Programs written in such languages can be understood to obey these rules by construction. The paradigm has predominantly been developed in the research area known as *language-based security*[41]. The downside to such techniques is the restriction to or dependency on specific languages and their concomitant software environments including development environments, debuggers, compiler infrastructures, and libraries and frameworks. Some relatively light-weight forms of language-based security technologies have achieved industrial importance, the most prominent example probably being the Java Bytecode Verifier originally developed and promulgated by (then) Sun Microsystems back in the 1990’s, which transferred ideas from the academically developed theory of *type safety* into large-scale industrial practice. Higher-end technologies such as proof carrying code have been harder to push into practice, because they rely on highly expressive logical systems incurring high specification overhead and requiring complex logical algorithmic techniques that are often beyond industrial scope. Language-based techniques have notably been used for ensuring *information-flow security* (see Taxonomy: Information flow methods), which appears to be directly relevant for certifying advanced properties such as data privacy. Other, related, directions of interest to compliance include *certified compilation* in the area of compiler verification. A notable long-ranging research project here is the project *CompCert*[42].

[40] DSL = Domain Specific Language: a computer language specialized to a specific application domain. https://en.wikipedia.org/wiki/Domain-specific_language
[41] https://en.wikipedia.org/wiki/Language-based_security
[42] https://compcert.org/

Logic-based methods

Logic-based methods are the only known methods that can in principle lead to actual verification of software systems, usually involving formal proofs of program properties. They can cover infinite (unbounded) behaviours and infinite (unbounded) data structures. They cannot be fully automated, but significant progress has been made in later years regarding their scope and applicability (a spectacular modern development is the verification of complex mathematical results using the Coq proof assistant[43]). Formal proofs are an ultimate form of verifiable certificate (checking a given formal proof can be done automatically, it is finding a proof that is hard). The downside to these methods is that they are still very difficult and costly to apply in general contexts outside of highly specialised application areas like, e.g., verification of cryptographic protocol implementations. Those very specialised areas may however be of some interest to Gaia-X Compliance. For example, concepts of Trusted Components (see Taxonomy: Architecture-based methods) might benefit from verification of certain specialised components. Although complete automation is impossible, practically interesting advances have been made in recent times (again, Coq is one of the leading systems). Most language-based methods can be understood as restricted logical techniques allowing for higher degrees of automation and ease of use. Logical attestation[44] is an example of a logical approach to attestation.

[43] https://coq.inria.fr/

[44] Sirer, Emin Gün, et al. *Logical attestation: an authorization architecture for trustworthy computing*. Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles. 2011.

Conclusion

Automation is needed

Regulation with respect to digital sovereignty is increasing at rapid pace in response to societal concerns that are central to European values and to Gaia-X. Increasing levels of regulation lead to the need for corresponding procedures for achieving and enforcing regulatory compliance (ex-ante or ex-post). Automation of compliance has inherent technical limitations, and compliance is embedded in a societal context which essentially involves the human and political factor (for example, jurisprudence). Still, automation *as far as possible* is needed for reasons of trust, scalability, and efficiency.

The need for automation of compliance procedures grows both with the volume and complexity of regulation and with the ever increasing complexity of systems subject to regulation. Compliance automation may benefit all stakeholders. From the courts' and the regulators' perspective, it is a reasonable concern that achieving and enforcing compliance of systems with regulatory policy may become ineffective or unrealistic, unless corresponding levels of automation of compliance (ex-ante and ex-post) are reached. From the perspective of providers of systems, it is a concern that providing systems to an increasingly regulated market place becomes increasingly difficult, unless corresponding tools for achieving compliance are accessible.

Part of the effort of automated compliance is to understand and as far as possible to specify which regulations are covered to which degree by specific compliance tools. Possible gaps between regulations and compliance procedures and tools should be identified as far as possible.

R&D towards automated compliance is needed

Just as security is by now a recognised area of research and development in computer science and related fields, the field of compliance tools and algorithms needs to be seen as a strategic subject of research and development to help fill the gap between regulation and systems subject to regulation. Currently, the gap between R&D resources invested in the creation of systems in need of compliance assessment on the one hand, and R&D resources invested in the creation of tools for achieving or enforcing compliance on the other hand, is disproportionate. The gap between the foreseeable amount of regulation on the one hand, and the R&D resources available to increase both understanding and automation of their implications for compliance, is becoming disproportionate.

Legal implications should be specified

It is up to the regulator to decide which tools may be used for compliance and how. Algorithmic compliance procedures and processes should produce *evidence* of *legal relevance* in assessing whether a given system is compliant with a set of rules. The legal implications of evidence produced by such procedures should be clarified *a priori* so far as possible. Relevance may both pertain to ex-ante properties and ex-post enforcement. Relevance may pertain to multiple stakeholders, including courts and judges, contract- and sla-management, and citizens. Furthermore, the legal implications and contractual circumstances of compliance tools provided by Gaia-X should be understood, for example with regard to legal commitment, responsibility, and liability.

Automation is needed for Labels

Gaia-X Labels constitute a (mostly ex-ante) instrument for creating levels of compliance and certification. Labels are distinct from the core notion of compliance given by the Gaia-X Trust Framework. Labels may go beyond the common standardised core of compliance regulation (such as found in the Gaia-X Trust Framework) at any given time, and the Label system and corresponding levels of compliance and certification may develop over time. The degree of automation associated with a Label may develop over time, for instance, as a result of new compliance technology being invented or implemented. But Labels should always be associated with formally stated requirements and should be subjected to automated compliance checking so far as possible.