bitkom

Bitkom Position Paper EU Data Act Proposal

Chapters 6 & 8

Switching & Interoperability

July 13, 2022

EXECUTIVE SUMMARY

We have to focus on significant obstacles and be realistic.

Switching cloud / data processing services seamlessly from an original service provider to a receiving can be a key requirement for certain customers in their respective situations. In principle, we support the Commission's objective to remove barriers for cloud-switching, which will help cloud service users to overcome vendor lock-in and dependencies on a single service provider if/where such is present, and to drive down costs of adopting a multi-cloud approach. However, the technical and operational complexities of migrating data, applications and workloads should be taken into account when enacting such policies. Like every standard, a right to switching is to be balanced meticulously with incentives for innovation and competition for all parties involved, as well as technical possibilities and impossibilities. It is impossible to draw a robust line between technical details and obstacles – every difference could count as an obstacle, so their significance is key. Also, technically, original service providers are not always able to ensure functional equivalence after a customer switched to a receiving service provider.

Complex projects require individual switching solutions.

Typically, cloud services (be it IaaS, PaaS, SaaS, XaaS) come with specific characteristics:

- Architecture complexity
- Project timeline
- Pricing model.

These characteristics exert enormous influence on the viability of switching rights / obligations in different settings. If they are highly complex, switching provisions should be negotiable between service provider and customer. This would avoid making certain ways of doing business impossible or unnecessarily difficult for both customers and service providers.

Customers and service providers need orientation.

Given the different ways of standardization the Data Act envisages for data processing services together with activities such as the Cloud Rulebook, we would very much welcome a comprehensive roadmap of the intended standardization activities to understand better how this would all fit together.
We would appreciate further clarity on many of the used definitions, e.g., the term “digital assets”, and would encourage relying on established terminology of the international cloud standard ecosystem instead of creating new terminology.

CHAPTER 6 – SWITCHING BETWEEN DATA PROCESSING SERVICES

Ch. 1., Art. 2, pt. 13 (-> Definitions)

"'service type' means a set of data processing services that share the same primary objective and basic data processing service model;"

- Neither "service model" nor the specification of a "basic data processing service model" are defined in the text. Furthermore, the term "service model" is not consistently defined (if at all) in the information technology or software engineering domains and may even be considered to include the business model-type of service configurations (such as in the ITIL standard regarding IT infrastructure services).
- Given the centrality of the term "service type" for the (non-waivable) possibility of switching service providers, this would deserve specification in sufficient detail to render the regulation effective and applicable.
- Furthermore, from an XaaS technical implementation point of view, conformity to the same service model (understood as an abstraction of some sorts of how a service is technically provided) usually is insufficient to allow porting data, digital assets, or applications as mandated by Art 23 para 1 point c.
- Technically, one needs identical or consistently compatible *actual* deployment and component architectures down to data formats and programming languages to ensure portability.

Ch. 6, Art. 23, para 1

"In particular, providers of data processing service shall remove commercial, technical, contractual and organisational obstacles, which inhibit customers from:"

- **It is unclear what is covered under a data processing service and would further welcome examples to illustrate this further, including how these obligations would apply to re-sellers and providers of managed hosting services.**
- **Regarding the definition of obstacles, we would appreciate further clarity what this would encompass as well as what this would not encompass. In particular, any configuration, communication, detail of any kind, could be considered an obstacle to switching, even the process to allow for switching could be considered an obstacle depending on its requirements.**
- **This cannot be the goal of the provisions and would create enormous legal uncertainty until there is jurisprudence. In order to sharpen the scope, increase legal certainty and avoid "false-positive" obstacles, we suggest the following phrasing for Art. 23 para 1:**

> [proposed definition] "In particular, providers of data processing service shall remove commercial, technical, contractual and organisational obstacles, which **significantly** inhibit customers from: […]"

- **Following this, details of what constitutes a significant obstacle could then be developed in standardization activities.**

Berlin,
July 13, 2022

Bitkom e.V.

David Schönwerth
Policy Officer Data Economy
T +49 30 27576-179
d.schoenwerth@bitkom.org

Lukas Klingholz.
Head of Cloud & AI
T +49 30 27576-101
l.klingholz@bitkom.org

David Adams
EU Public Policy Officer
M +32 471 927890
d.adams@bitkom.org

Albrechtstraße 10
10117 Berlin
Germany

President
Achim Berg

CEO
Dr. Bernhard Rohleder

- **We suggest adding that 1°) whoever complies with standards under Art. 26 para 3 would be assumed in compliance with Art. 23 para 1, to the extent such standards cover the obligations in Art. 23 para. 1; and 2°) apply this to IaaS as well, ie extend Art. 26 para 3 to all XaaS, not only PaaS and SaaS. 1. This will create legal certainty while also providing further incentives to fulfilling Art. 26 standards, that is creating a harmonized landscape. Here, the passus in Article 40 of the AI Act Proposal could be an inspiration.**[1]

Ch. 6, Art. 23, para 1 pt. (a)

"(a) terminating, after a maximum notice period of 30 calendar days, the contractual agreement of the service"

- We kindly offer the following background information before suggesting clarifications to balance them with the idea of enhanced cloud switching.
- A maximum termination period of 30 days (which could hence theoretically come down to 1 day) restricts the freedom of contract of both, data processing providers and their customers. If such a short termination period was in place, data processing providers would not be able to give any discounts on long-term commitments of customers which would increase costs for all customers even those willing into enter long-term relationships.
- Such an obligatory provision would also limit the effectiveness and reach of any partnership, data space, or other form of ecosystem, where enterprises and organizations typically may want to enter and honor long-term relationships without the possibility to terminate with such a short notice period. As a matter of fact, it would exclude all enterprises of the European XaaS community from effectively contracting longer-term contracts. This would create severe economic challenges for the XaaS market.
- **To properly balance this with customers' switching rights, we suggest to carefully evaluate the notice period against different service types, as the need for a short notice period depends on certain characteristics of each (cloud) project. As mentioned at the top, these include in particular:**
  - **Architecture complexity**
  - **Project timeline**
  - **Pricing model.**
- **Furthermore, there is an important difference between the contractual switching timeline and the technical switching timeline. In other words, while terminating a contract can be rather straightforward, porting a complex enterprise project to a receiving service provider can be virtually impossible within 30 days, depending on the circumstances (see above too).**
- **For the technical switching timeline, we would suggest gathering further evidence of how this works in different scenarios. We foresee that in some, 30 days would be a reasonable technical switching timeline (e.g., moving certain database content), while in others, this**

[1] „High-risk AI systems which are in conformity with harmonised standards or parts thereof the references of which have been published in the Official Journal of the European Union shall be presumed to be in conformity with the requirements set out in Chapter 2 of this Title, to the extent those standards cover those requirements.

would be completely unprecedented due to its impossibility under any reasonable setting of staff and resources (e.g., moving full integrated enterprise IoT platform).

- As an example, a managed-service contract will typically require longer switching times than a pay-as-you-go offer for ad-hoc use of an SaaS offer.
- **Therefore, we suggest removing the reference to the maximal period of 6 months in Art. 24.2, because if a service provider can prove the technical unfeasibility, customers and service providers should be able to prolong the switching period for as long as necessary to make it technically feasible.**

Ch. 6, Art. 23, para 1 pt. (c)

"(c) porting its data, applications and other digital assets to another provider of data processing services"

- It is unclear what "porting" exactly means from a technical point of view, especially for data and for applications. For instance, data may be represented in various syntactical forms or conform to different semantical specifications. Does "removing obstacles of porting" mean that data has to be provided in the original format (as natively used by the original service provider) or does the original service provider have to further assist the customer in porting, e.g., by providing mapping logic from its data format to the data format used by the receiving service provider.
- In a cloud-native PaaS/SaaS environment, the term "application" often no longer makes sense because PaaS/SaaS providers regularly do not provide "applications" in the sense that one or more executable files are specifically deployed to a computation node and execute on behalf of a user (especially in the case of service offerings capable of multi-tenancy). Instead, all cloud service providers today offer so-called capabilities the provision of which is spread over a distributed and elastic computing infrastructure (e.g., containers, but also load balancers, IAM systems, API management side cars, monitoring probes, etc., database and other backend services) involving a plethora of classical "applications" to provide it.

  As a matter of fact, porting "applications" to a receiving service provider can hardly be achieved economically in any PaaS/SaaS setting unless the two service providers use the very same cloud architecture. It is certainly insufficient for the two service providers to share the "basic data processing model" as referred to in the definition of "service types" included in Art. 2(13), because the actual component architecture matters here, not just the processing model.
- **In-as-much as porting is concerned, we suggest substituting the term "application" by the new term "functional logic" with the meaning of a connected set of data (including any metadata) capable of influencing, amending, or changing the behavior of a data processing service after proper deployment.**
- To the extent that customers of data processing services are able to generate "functional logic" on their own (e.g., defining rules for a rules engine capability, defining mappings or transformations of data pipelines, defining specific microservices, e.g., Docker containers using data processing service-specific APIs, able to execute in the context of the data processing service), it is unclear what "porting" this can mean – as typically the underlying technologies of data processing services will vary in incompatible ways to allow any simple form of porting

- In light of the fact that all a customer's "digital assets" have to be portable to a receiving service provider, the scope of the term "digital asset" is decidedly too broad when it includes every single piece of data, application, or any other digital element for which the customer merely has the right of use (Recital 72). In a PaaS/SaaS setting, customers typically have access to and the right to use for many different but generic pieces of functionality such as dashboards and widgets, identity and access management systems, role-based access controls, workflow engines, reporting engines, search & query mechanisms, rules engines, messaging and eventing mechanisms, storage functionalities, and many more. Making the original service provider responsible for porting all of this to a receiving service provider (this seems to be the current proposal) essentially requires that the whole PaaS/SaaS offering needs to be portable to a receiving service provider irrespective of any IP rights and any architectural differences of the services provided.
- Furthermore, the term "digital assets" is not defined in the Data Act body but only in its recitals. In addition, this term does not seem well-defined neither well-established in the international cloud standard ecosystem to the best of our knowledge.
- **We suggest scoping this Article to "data" and "applications" while refraining from including digital assets.**
- **Generally, we suggest referring to European/international cloud standards, rather than creating new terms as they would have to be developed, understood, and aligned with existing standards.**
- **We further suggest considering trade secrets, patents as well as other IP-related rights in the context of data and digital assets, as such could offer significant insight for competitors and other actors into competitively sensitive details of cloud services.**

Ch. 6, Art. 23, para 1 pt. (d)

"(d) maintaining functional equivalence of the service in the IT-environment of the different provider or providers of data processing services covering the same service type, in accordance with Article 26"

- **What exactly are (technical) obstacles inhibiting customers from maintaining functional equivalence at the receiving service provider(s)? Following items (a) to (c), the customer has already ported all its data, applications, and other data assets (whatever these may be) to the receiving service provider: What other technical obstacles would remain at the side of the original service provider? The only things remaining are internal characteristics of how the original service provider essentially realized its data processing services – very likely a trade secret in any case.**

Ch. 6, Art. 23, para 2

"2. Paragraph 1 shall only apply to obstacles that are related to the services, contractual agreements or commercial practices provided by the original provider"

- The task of porting digital assets (including data and applications here) from an original service provider to a receiving service provider **can hardly be the sole responsibility of the original service provider alone** as the cooperation of the receiving service provider is always and invariably required over the whole course of the migration project.
- As currently formulated, the situation can arise where the original service provider has indeed removed all obstacles of porting a customer's digital assets to a receiving service provider (as required by Art 23 para 1) but that obstacles in the sphere of the receiving

service provider prevent the customer from actually switching service providers. In such a case, the customer does not have any legal redress out of this regulation for remedying the situation and they will, consequently, not be able to effectively switch service providers.

- To address this, the **receiving service provider should have an obligation to support and co-operate**. This is instrumental for a successful process. Probably, the draft obliges the original service provider to establish clear obligations and make it enforceable. However, to overcome the fact that the receiving service provider also has something to do, a specification that details the switching process would help. This is how the telecommunications industry developed porting.

Ch. 6, Art. 24, para 1 pt. (a)

"(a) clauses allowing the customer, upon request, to switch to a data processing service offered by another provider of data processing service or to port all data, applications and digital assets generated directly or indirectly by the customer to an on-premise system"

- **We acknowledge that the need to port data from one service provider to another is of paramount importance for a customer's capability to effectively switch service providers. However, in certain situations – such as IoT/IIoT/Industry 4.0 – the sheer amount of all data generated by the user (i.e., its devices or machines) over the course of the usage of a data processing service may be significant and thus too large to be continuously stored by the original service provider to be available upon switching, especially since Art. 24(1) requires all data "generated directly and indirectly" to be ported by the original service provider. Putting a disproportionate burden on the original service provider in edge cases could lead to prohibitively high cost/effort, if on-premise** porting is possible at all, that could clear certain offers from the market if service providers cannot comply. To avoid this, we suggest navigating this situation depending on these characteristics:
  - Architecture complexity
  - Functional complexity
  - Amount of data.
- **Where these three characteristics are highly developed, we suggest giving more flexibility to agreeing on differing terms between the parties involved, including but not limited to the cost involved.**
- **The Data Act should take into account the technical prerequisites and practical consequences of exercising such clauses, thus we suggest adding safeguards against abusive invocations of such rights.**

Ch. 6, Art. 24, para 1 pt. (a)

"(a) clauses allowing the customer, upon request, to switch to a data processing service offered by another provider of data processing service […], in particular the establishment of a mandatory maximum transition period of 30 calendar days, during which the data processing service provider shall:
(1) assist and, where technically feasible, complete the switching process;
(2) ensure full continuity in the provision of the respective functions or services."

- It seems unfair that the source service provider is obliged (without any limits) to assist the customer but the receiving service provider has no such obligation, not even an obligation to cooperate.

- In any non-trivial two-party data migration scenario (like moving from one XML editor to another editor for the same version of the XML standard) it is unrealistic if not impossible to expect that the original service provider may alone ensure "full continuity" of the service provision without any assistance by the customer (upon whose request this migration is being performed), and notably without any assistance or even cooperation obligation by the receiving service provider.
- Even in traditional outsourcing contracts, which are subject to lengthy negotiations, clients/users and service providers agree on specific service level agreements that service providers must comply with during the termination assistance phase, where service providers help clients to migrate their workloads to a receiving service provider. The service levels agreed therein never foresee a 100% service continuity, as parties understand and agree that the service will not be the same during a termination phase than during the lifecycle of the contract. Parties also know that business and service continuity is better guaranteed through collaboration between the service providers (both original and receiving services provider) and the client, rather than through shifting obligations on the original service provider.
- The formulation of item (2) does not seem to consider how non-trivial data or application migration projects work, by presuming that the customer can enjoy an uninterrupted (arg. "full continuity") provision of a given service while migrating. There is either a hard cut-over date and time, where users have to switch from an original service provider to a receiving service provider (this is discontinuous) or there is a kind of mediation gateway on top of both, i.e., the original service and the receiving service in parallel. Depending on the state of the migration (or porting), this mediation gateway dynamically determines whether to route a service request by a user to the old instance of the service at the original service provider or to the new instance of the service at the receiving service provider. This, however, is not conducted in the form of simply "switching" but needs a full-fledged migration project to be set up (and paid) by someone.

Ch. 6, Art. 25

- The Commission's analysis shows the importance of collaboration[2] and interoperability between service providers to facilitate low-cost switching between cloud service providers. We welcome the proposed gradual withdrawal of switching charges, which constitute an important obstacle to cloud-switching and adopting multi-cloud approaches. However, to provide more clarity, the term "charges" and the notion of "costs directly related to the switching process" should be defined in the legislative text.

Ch. 6, Art. 26, para 1

"Providers of data processing services that concern scalable and elastic computing resources limited to infrastructural elements [...] shall ensure that the customer, after switching to a service covering the same service type offered by a different provider of data processing services, **enjoys functional equivalence** in the use of the new service."

- **We are unsure how such post-switching functional equivalence could be ensured by the original service provider. It seems extremely difficult to impossible for the original service**

[2] Impact Assessment Report EU Commission, 2022

provider to ensure functional equivalence of the respective service during or even more so after the completion of the switching procedure.

- We believe that to achieve interoperability in the IaaS sector, (i) the scope of any potential functional equivalence requirements should be agreed with the customer in the contract; (ii) interoperability and any functional equivalence related thereto should be a joint task for both the original and the receiving service provider; and (iii) service providers should not be performing work outside their environment. Also, we think that IaaS, like SaaS and PaaS, could also be subject to interoperability standards identified in accordance with Article 29.5, to demonstrate compliance with Article 26 and the technical aspects of switching.
- Also, as the obligation of the original service provider is **not limited in duration,** it would imply that the original service provider would have to provide an eternal and universal guarantee and insurance of its former customer regarding the rendering of services by the receiving service provider. Such a requirement seems rather unwarranted.

Ch. 6, Art. 26, para 2

"For data processing services other than those covered by paragraph 1, providers of data processing services shall make open interfaces publicly available and free of charge"

- We would greatly welcome more clarity what such open interfaces would cover.
- We fail to see any reason why these open interfaces should be availably publicly, which (in its normal semantics) means to the general public, i.e., anyone. This should be reduced to active or former customers of the original service provider (potentially the receiving service provider as well) and only until the expiration of any termination or migration period.

Ch. 6, Art. 26, para 3

"[PaaS/SaaS] providers of data processing services shall ensure compatibility with open interoperability specifications or European standards for interoperability that are identified in accordance with Article 29(5) of this Regulation."

- As currently no such standards exist, any service provider would have to change their services and APIs in case any such standard entered into force. Any such implementation, however, needs a stable standard and sufficient time on the side of the service providers before the service providers can, with the required professional duty of care, declare compatibility to such an interoperability standard.
- In the current formulation, however, service providers are supposed **to be capable of immediately and instantaneously switching to** a completely new (interoperability) standard immediately after any such standard comes into force. This cannot be realistically achieved. Hence, we propose to allow service providers a reasonable period of time for implementing any such standard.

CHAPTER 8 – INTEROPERABILITY

Ch. 8, Art. 28, para 1

"1. Operators of data spaces shall comply with, the following essential requirements to facilitate interoperability of data, data sharing mechanisms and services"

- The scope of Article 28 remains unclear as definitions of a "data space" and operators of such are missing. This would include but not be limited to Data intermediation services as defined in the Data Governance Act), and Common European Data Spaces.
  This will also have to include a demarcation from and a clear contrast vis-a-vis Gaia-X and similar workstreams, such as those under the Digital Europe Program, as well as other existing or future data platforms and data sharing platforms.

  > We propose a definition along the following lines:
  > "A data space is a coordinated set of technical standards, organizational policies, and data space services under a specified governance model to enable and facilitate data exchange between its participants."

- Usage of the term "operator" seems to hinge on a model of a single (centralized) data space where a single legal entity is responsible for the whole data space and its data space ecosystem services. However, clarity would be needed between the allocation of responsibilities under this provision to match with a practical division of competences between operating companies and governing bodies, which may coexist in data spaces.
- It is further unclear how this provision would interact precisely with **federated data spaces** with more than one operator (such as the Gaia-X initiative has been discussing for some time) and also **decentralized data spaces** (e.g., using distributed ledger technology in the form of so-called DAOs, decentralized autonomous organizations.

Ch. 8, Art. 28, para 1 pts. (a)-(c)

"(a) the dataset content, use restrictions, licences, data collection methodology, data quality and uncertainty shall be sufficiently described to allow the recipient to find, access and use the data;
(b) the data structures, data formats, vocabularies, classification schemes, taxonomies and code lists shall be described in a publicly available and consistent manner;
(c) the technical means to access the data, such as application programming interfaces, and their terms of use and quality of service shall be sufficiently described to enable automatic access and transmission of data between parties, including continuously or in real-time in a machine-readable format;"
(d) "the means to enable the interoperability of smart contracts within their services and activities shall be provided."

- The points (a)-(c) do not seem to acknowledge that most of the documentation requirements have to be provided by the actual participants of a data space. Any operator or governing body (including federated or decentralized forms) will not be able to infer or determine this information by themselves.
- We also like to stress the fact (potentially in a suitable recital) that many of the pieces of information constituting the data space operator's documentation obligations cannot be met by the data space operator but have to be provided by the data provider. This,

essentially, means that the operator can only roll over these obligations (e.g., by means of a suitable data space participation agreement) to the actual data space participant providing the data.

- Regarding provision (c), we are unsure about the allocation of responsibilities between the data space operator as facilitator of data exchange, on the one hand, and actual data space participants as persons responsible for allowing and technically enacting the data exchange, on the other. This provision seems to build on the idea of holding the data space operator accountable for the actual data exchange (i.e., quality of service, real-time data exchange capabilities) when, as a matter of construction, the provision of any and all APIs regarding to data exchange could be in the sole sphere of influence of the data space participants.

Ch. 8, Art. 28, para 1 pt. (d)

“the means to enable the interoperability of smart contracts within their services and activities shall be provided.”

- We understand that the only hard requirements (in the sense that real implementations need to realize the requirement instead of just being able to provide a documentation or description) is with regard to smart contracts which neither are at the center nor needed in many data space conceptualizations and architectures. We are also surprised that in this provision all smart contracts used in the data space are included as opposed to the much narrower regime of Art 30 para 1 where only smart contracts “in the context of an agreement to make data available” are subsumed.
- We note that the interoperability requirements of Art. 28 para 1 pt. d– if needed – are strictly limited to all smart contracts used in a data space and do not also extend to any other data exchange logic e.g., also provided by classical applications or other cloud or on-premise data processing services.
- The current formulation seems to disincentivize the use of smart contracts because of the interoperability requirements – whereas any standard programmed business logic used in the data space (e.g., policy enforcement, logging) does not need to be interoperable. At the same time, we would consider including all of such to be completely unwarranted given their early level of maturity. Thus, we suggest moving smart contracts out of scope of Art. 28 para 1 pt. d and reconsidering the need for Art. 30. As a second option, we suggest aligning the scope of Art. 28 to the narrower scope of Art. 30 para 1.
- It remains unclear what “interoperability of smart contracts” should concretely mean. The standard definition of interoperability (= the ability of software to exchange information and make use of it) is too wide and unspecific because it is tantamount to requiring that every program of a certain type (viz., smart contracts executing on a DLT) needs to be interoperable with every other program. Until a standard has been really adopted, data space operators and participants alike do not have any guidance at all regarding this obligation. This will stifle innovation as market participants cannot ensure *ex ante* that their investments will conform to future (currently unknown) standards potentially requiring huge re-work or adaptations (of the smart contracts).
- It remains unspecified whose smart contracts this provision is referring to: Are these smart contracts used by the data space operator itself (like for authentication) or are these smart contracts used by the participants (like “compute to data”) or provided by third parties? We

further are unsure about the technical substrate for limiting "interoperability of smart contracts" as pertaining to any "services and activities" rendered by data space operators. This simply can mean everything or nothing and needs to be specified more thoroughly before entering into force to prevent uncertainty for all data space actors.

Ch. 8, Art. 28, para 4

„4. The Commission may, in accordance with Article 10 of Regulation (EU) No 1025/2012, request one or more European standardisation organisations to draft harmonised standards that satisfy the essential requirements under paragraph 1 of this Article"

- We suggest indicating (potentially in a suitable recital) potential existing standards that can be thought of as (precursors to the) "harmonized standards". This would give data space operators some guidance of the direction the EC is thinking w.r.t. to this important piece of currently not fully available standardization.
- The Commission should issue standardization requests to specify the technical details for the essential requirement to the European Standardization Organisations at an early stage.

Ch. 8, Art. 28, para 5

5. The Commission shall, by way of implementing acts, adopt common specifications, where harmonised standards referred to in paragraph 4 of this Article do not exist or in case it considers that the relevant harmonised standards are insufficient to ensure conformity with the essential requirements in paragraph 1 of this Article, where necessary, with respect to any or all of the requirements laid down in paragraph 1 of this Article. Those implementing acts shall be adopted in accordance with the examination procedure referred to in Article 39(2).

- In principle, the European Commission should make use of harmonized standards or open interoperability specifications to specify the technical details for the essential requirements. The instrument of "common specifications" must remain a fallback option. The wording of the Commission's proposal does not reflect this and should therefore be supplemented by concrete, narrowly defined requirements for the adoption of common specifications.

Ch. 8, Art. 29, para 2

"2. Open interoperability specifications and European standards for the interoperability of data processing services shall address:

(a) the cloud interoperability aspects of transport interoperability, syntactic interoperability, semantic data interoperability, behavioural interoperability and policy interoperability;

(b) the cloud data portability aspects of data syntactic portability, data semantic portability and data policy portability;

(c) the cloud application aspects of application syntactic portability, application instruction portability, application metadata portability, application behaviour portability and application policy port."

- The characterizations of "cloud interoperability" given in point a make heavily use of a concept of "interoperability" which is not aligned with the definition given in Art 2 pt. 19 as they much more (and rightfully so) relate to the concept of portability (instead of interoperability). Hence, we suggest adding a separate definition of “portability” (see this commentary above).

- point a “syntactic interoperability”: With a look to the following term, “semantic data interoperability”, we think this should read “syntactic **data** interoperability” (adding the qualifier “data”).
- Points b and c both (rightfully) center around the term “portability” which is not defined anywhere in the proposal. Hence, we strongly suggest defining this pivotal term in Art 2 (see our proposed definition above). // Systematically replace interoperability with cloud portability.
- Following the pattern of point b, we believe point (c) should correctly be called “the cloud application **portability** aspects […]” explicitly mentioning, like point (b), that we are talking about porting cloud applications.
- We believe the attempt to characterize cloud application portability in the five dimensions of point (c) does not align enough with software engineering practices or research. It is unclear to us what “application syntax” exactly refers to. The programming language(s) an application is written in? The syntactical (= data format related) properties of an application’s API? What is “application instruction portability”? Are these the commands (like in a spreadsheet or rules engine) users can use to work *within* the applications or are these the instructions of the underlying programming language the application is written *in*? In today’s API-led economy this should rather point to the compatibility of the externally exposed APIs of an application. We suggest using a goal-oriented definition of “application portability” like suggested by use in our commentary to Art 2 pt. 19.

Ch. 1., Art. 2, pt. 17 (-> Definitions)

“’electronic ledger’ means an electronic ledger within the meaning of Article 3, point (53), of Regulation (EU) No 910/2014”

- The reference resolves to the following definition of «electronic ledger» (note that this proposal has not entered into force yet):

> “(53) ‘electronic ledger’ means a tamper proof electronic record of data, providing authenticity and integrity of the data it contains, accuracy of their date and time, and of their chronological ordering;”

- This definition is wide to the extent that every existing database (irrespective whether a relational database or NoSQL) supporting (i) ACID transactions and (ii) a change log (recording all the changes to the database) would fall under it. The definition also does not take into account the characteristics of distributed ledger technologies (e.g., blockchains, directed acyclic graphs, and other decentralized systems for establishing consensus on a common shared state) – see later in this section for a suggested definition capturing the essential capabilities of DLT.
- In combination with the definition of a «smart contract» as per Art 2 pt. 17, every stored procedure of such a database would count as a smart contract.
- We believe that these wide definitions lead to unintended consequences in conjunction with Art 30 where every stored procedure used “in the context of an agreement to make data available” (e.g., for access control, verification of usage rights etc.) is subject to the essential requirements and to the EU conformance testing. This will negatively affect every

existing PaaS/SaaS vendor or service provider who uses stored procedures on their databases when making available the data of their platforms. All these CSPs will have to certify their "smart contracts" (= stored procedures"). Additionally, ensuring the essential requirements of Art 30 for every stored procedure seems to be vastly disproportionate to the regime the Data Act wants to regulate. Stored procedures have been extensively and widely used for decades now without anyone ever demanding them to fulfill the "essential requirements" of Art 30 para 1.

- Furthermore, we note that the Data Act uses the term «electronic ledger» whereas both MiCA (Markets in Crypto Assets) and TFR (Transfer of Funds Regulation) proposals use «distributed ledger technology» (DLT):

> MiCA Art 3 pt. 1 specifies
> "'distributed ledger technology' or 'DLT' means a type of technology that support the distributed recording of encrypted data;"

- However, also this definition (TFR simply refers or reuses the definition provided by MiCA) fails to capture the essential features of DLT and is far too broad and unspecific. Again, any distributed database (e.g., if you have a cluster of two database instances for ensuring high availability and disaster tolerance) will be regarded an «electronic ledger» in MiCA and TFR. We also fail to recognize the requirement that the database needs to support storing of "encrypted data" as virtually every database supports that: If you enter encrypted data, it will be stored. Encryption is also an ephemeral (if any at all) characteristic of a DLT.
- As a remedy, we propose to consistently use the term "distributed ledger technology" as MiCA and TFR do and define it as follows using «distributed ledger system» first.

> [proposed definition] "A «distributed ledger system» is a software system consisting of distributed subsystems collectively executing algorithms allowing to create and to distribute certain atomic digital assets between these subsystems essentially relying on cryptographic means to ensure authenticity, integrity, security, immutability, and consistency of its operations. "
>
> [proposed definition] "'Distributed ledger technology' (DLT) means a type of technology directly supporting the implementation of a distributed ledger system."

- We believe that this narrower definition is excellently able to capture the essence of DLTs at the core of where the Data Act should regulate smart contracts. Existing databases and stored procedures – even when used for sharing data – remain unaffected as this type of usage patterns have not created any need for the essential requirements stated in Art 30(1).

Ch. 8, Art. 30

- We are rather doubtful about the necessity of Article 30 as general IT security and other established technical standards regarding the development of software/applications (smart contracts are applications) apply already.
- In particular, this would severely challenge existing business models and processes relying on smart contracts in data sharing settings that are working well. In particular, issuing an EU conformity declaration for such would significantly hinder innovation. Against this background, as a second-best option only, we would like to offer the following suggestions:

Ch. 8, Art. 30, para 1, pts. (b)-(d)

(b) safe termination and interruption: ensure that a mechanism exists to terminate the continued execution of transactions: the smart contract shall include internal functions which can reset or instruct the contract to stop or interrupt the operation to avoid future (accidental) executions;
(c) data archiving and continuity: foresee, if a smart contract must be terminated or deactivated, a possibility to archive transactional data, the smart contract logic and code to keep the record of the operations performed on the data in the past (auditability); and
(d) access control: a smart contract shall be protected through rigorous access control mechanisms at the governance and smart contract layers.

- It remains unspecified who is allowed or obligated to perform the functions mentioned in (b)-(d). For instance,
  - who needs to "foresee" that a smart contract needs to be terminated?
  - Who needs to activate the "data archiving" function?
  - Who needs to provide the storage for archiving transaction data? For how long? This also applies to cases where this information is stored on the distributed ledger because many ledgers allow the pruning of the transaction history to save storage space.
  - Is it possible to roll over the archiving requirements to the participants of a data space?
- What is the "**governance layer**" referred to in point d? Does that refer to the technical deployment process of smart contracts or is that an organization process (e.g., when on-boarding new data space participants and they all get a unique ID or similar processes)?
- Currently, we are not aware of a comprehensive and exhaustive European specification **how to make smart contracts legally binding** upon the parties participating in their execution. Without such an overarching "governance and trust layer", we fear, that the actual usage and proliferation of smart contracts will be delayed. For instance, which form of identity may be safely used in a smart contract (eIDAS, EDI, Gaia-X DIDs, X.509 certificates), how can we reliable ascertain consent to a (smart) contract (who is the offeror, who the offeree?), how is liability distributed in a smart contracting environment because we always have a DLT as underlying execution engine typically operated by a plethora of other actors (validators, node operators, consensus committees, etc.)?

Ch. 8, Art. 30, para 6

"Where harmonised standards referred to in paragraph 4 of this Article do not exist or where the Commission considers that the relevant harmonised standards are insufficient to ensure conformity with the essential requirements in paragraph 1 of this Article in a cross-border context, the Commission may, by way of implementing acts, adopt common specifications in respect of the essential requirements set out in paragraph 1 of this Article. Those implementing acts shall be adopted in accordance with the examination procedure referred to in Article 39(2)."

- In principle, the European Commission should make use of harmonized standards or open interoperability specifications to specify the technical details for the essential requirements. The instrument of "common specifications" must remain a fallback option. The wording of the Commission's proposal does not reflect this and should therefore be supplemented by concrete, narrowly defined requirements for the adoption of common specifications.