

Building a public mobile stack

A framework to align European mobile stack alternatives with user values and EU laws

Danny Lämmerhirt, Heleen Janssen, Isadora Dullaert, Etienne Valk, Iris Toepoel, Christian Cordoba Lenis

Table of contents

Glossary	7
1 Introduction	8
2 Our methods	19
3 User values and the mobile stack	22
4 EU Law applying to user values & the mobile stack	27
5 Conclusion	34
Literature	36
Colofon	38

Building a Public Mobile Stack.

A framework to align European mobile stack alternatives with user values and EU laws

Danny Lämmerhirt

Heleen Janssen,

Isadora Dullaert,

Etienne Valk,

Iris Toepoel

Christian Cordoba Lenis¹

With support from: Bente Zwankhuizen, Claudia van Swelm, Simone van der Burg,
Natalie Sahusilawani, Aris Ham, Marit Hoefsloot, Aris Ham, and Alain Otjens

© Waag Futurelab, September 2026

This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 License



¹ The team has contributed to this report in the following capacities: Danny Lämmerhirt (principal investigator for user value analysis) led the conceptualization, methodology development, and final production of the framework, and contributed to the empirical analysis of user values. Isadora Dullaert contributed to the conceptualization and methodology development, and led the empirical analysis of user values. Heleen Janssen (principal investigator for legal analysis) led the conceptualisation, methodology development and final production of the legal framework; Etienne Valk, Iris Toepoel and Christian Cordoba Lenis contributed by writing the legal report.

Executive summary

This report explains how alternative European mobile technology can better reflect user values and needs while complying with applicable EU law. The report is part of a broader effort to build European digital public infrastructure. Our central argument is that developing ethical mobile stack layers is not a neutral process: it requires working directly with user groups and grounding design choices in EU law. As mobile stack layers increasingly function as infrastructure, offering essential services to a wide range of users with different needs across societal sectors, broad policy goals or appeals to abstract ‘public values’ are not sufficient on their own to guide those developing the mobile stack. This report is the outcome of research conducted under Mobifree, a Horizon-funded innovation action project developing a human-centred and ethical mobile stack for European citizens and organisations.

Research goals and approach

The objective of this report is threefold: 1) to develop an analytical framework connecting user values to specific layers of the mobile stack and relevant design interventions, 2) to demonstrate the importance of involving users in software that is designed for them; and 3) to offer an analysis of relevant EU-based legal frameworks for the development and evaluation of compliant mobile stack components. Based on our research, we propose a framework built on two processes: aligning stack layers with user values, and ensuring compliance with EU law. Taken together, these processes provide a holistic approach to ethical software development. This research is based on workshops (n=102) and pilot testing (n=127) with participants across four user groups (young adults, civil servants, humanitarians, and mobile developers), and the analysis of fourteen EU laws that apply to activities of mobile ecosystem developers.²

Findings

- Tests with Mobifree software show that different **user groups appreciate many of the tested mobile stack alternatives** and the values they represent (e. g. privacy preserving features, transparency-enhancing features).
- **There is a gap between values as abstract concepts and how users perceive them in practice.** Understandability and usability are important factors to invest in to ensure that values built into stack layers are actionable for users and provides a clear added value for them.
- **Users prioritise different values** and can **understand values and associated software functionality in different ways.** Therefore, testing with different user groups is required and user values should be weighed against one another to limit potential harms for some user groups at the benefit of other users.

² Research data of the legal analysis as well as user values can be found as open data at Zenodo, available at: <https://zenodo.org/records/22918474> and here <https://zenodo.org/records/22918307>.

- **Mobile stack layers are interdependent and can enable or hinder value alignment and legal compliance.** Value alignment therefore requires a holistic view of the architecture of software to identify on which stack layer values should be implemented.
- Nearly all **user values** identified in the empirical research are already **reflected** in the **EU laws**, but these laws may not get a developer's attention they should. Regulatory guidance can more explicitly connect user values and user's legal rights with a developer's obligations to effectuate these rights, by a regulator's unambiguous explaining, and giving examples, of how law may shape a developer's design.
- Developing parties can refer to user values to shape the definition and interpretation of open-natured laws. For example, the way privacy is described in the law may not provide enough guidance for technology design. Developing parties could look at regulatory guidance materials and user values in order to find an appropriate approach to design for privacy.

Recommendations

We recommend that **policymakers** creating and implementing EU laws and policies:

- **Develop full-stack policy:** Policymakers should adopt a stack-based view of software governance. If dependencies between stack layers are not accounted for, interventions at one layer may leave underlying dependencies unaddressed. These dependencies can seriously hinder the development of alternative mobile software stack layers, and with this, innovation. Policy should therefore assess how decisions at one layer affect other layers of the stack and address dependencies across the system.
- **Build stack layers and ecosystems to avoid dependencies:** In cases where the EU develops its own stack layers as public infrastructure, it needs to ensure independence from underlying commercial stack layers. Many government-built technologies, such as identity services, currently rely on underlying big tech stack layers. This jeopardises EU sovereignty and user autonomy. Funding agencies can play a role in this by investing in ecosystem building, rather than single technologies.
- **Fund ethical technology research and development:** Policymakers should encourage developers to engage with users to assess how (public) values translate into specific contexts of use. Funding agencies should free up financial resources to support dedicated user research and development processes for ethical technologies, focused on aligning technologies with users values, needs, and EU laws. This also advances innovations around European stack alternatives that help foster digital autonomy of citizens and companies.

- **Creating a market for alternative mobile stacks:** Decision makers in organisations should provide incentives for public institutions and companies to switch to (free and) open source alternatives. This can include to establish of clear standards and policies as well as and procurement guidelines for the purchase and use of alternative mobile software stacks, as well as measures to support transitions, such as through training programs.
- **Introduce regulatory guidance for mobile stack developers** at detailed level about a law's interpretation and application, supported by examples across societal sectors (comparable to the EDPB's guidance on key GDPR's elements). This applies particularly whenever developers have to interpret open-natured, risk-based laws where developers themselves are tasked to assess a law's application to their own data processing intentions and data processing methods. Regulators should particularly pay attention to mobile stack developers that enter the market that usually do not fully focus on legal compliance, whereas they may offer the services and goods of tomorrow. Overall, making design compliant as of design stages is key to our shared safe and secure digital future. We strongly recommend regulators to design templates for risk impact assessments and for compliance assessment.

We recommend that **developers:**

- **Test whether their value-optimised technology is usable and comprehensible:** (FOSS) developers wanting to make their products more appealing to diverse user groups have to invest in user experience. This includes improving the aesthetics of interfaces and the understandability of features, for instance through more accessible language.
- **Build user research into iterative development cycles:** software development should include user research processes that take into account the needs and values of relevant stakeholder groups. This includes planning, building, and testing value alignment and defining user groups clearly that could be (negatively) impacted by your product.
- **Organise internal governance and decision-making to accommodate user perspectives:** Organisational structures of developer teams influence their ability to accommodate value-aligned insights, and shape decision-making responsibility. Developing parties should review and if needed adjust decision-making processes to ensure that insights from user feedback are taken into account.
- **Adopt a broader outlook on user values beyond values like freedom and openness of software, or privacy:** Privacy or freedom and openness of software do not reflect the full spectrum of ethical software and may not reflect what users find most important. It is essential to look at the values as expressed in the empirical research with users, as well as in the law.

Glossary

Digital public infrastructure: Digital public infrastructure is a container term that is applied to different types of digital infrastructure and whose public character can be defined in different ways. Increasingly, it is understood to be publicly controlled, to manifest public values in built characteristics (e.g. being open and interoperable), and to support collective goods (e. g. well-being and autonomy).

Infrastructure: We understand infrastructure as the material basis necessary to circulate goods and services and organise social activities. An infrastructure is robust, widely shared among different user groups, widely accessible, and essential. To become infrastructure, a system often relies on standards, user habituation, and routine use. Some infrastructures attain the status of a public good, such as utilities or railroad networks.

Mobile Stack: A stack describes the multiple, and connected technological components involved in computation. The stack draws on programming and information sciences and understands these components as layers. A mobile stack is a specific type of stack that usually includes layers such as apps, middleware, mobile operating systems, app distribution systems, hardware, and cloud compute.

Values: Values describe what is important in someone's life. Values are not preferences or opinions, but describe the principles by which people judge if something or someone is important, worthy, or of quality. Different users can find different values important when interacting with mobile software.

User values: User values refer to those values expressed by individual end-users, for instance when interacting with mobile software. In this report, we refer to user values to describe how users experience and assess free and open source mobile software.

Public values: Public values are those values that shall be applicable to the general public. In our report, we mention public values in relation to EU laws that enforce privacy, competition, and other public values.

1 Introduction

Over the past two decades, big tech companies including Apple, Google, Microsoft and Facebook have shaped the mobile software ecosystem. This has resulted in an ecosystem fuelled by opaque, extractive and exploitative business models relying on user data. What is more, the development of technical standard components requires both developers and consumers to obey the terms set by these monopolists. The power of large US tech companies in the mobile software market not only undermines Europe's competitive position and innovation more broadly, it also clashes with core European values such as transparency, respect for (digital) human rights or appropriate working conditions for employees.

A handful of big tech companies control different stack layers of mobile software and hardware. A stack describes the multiple and connected technological components - also known as stack layers - involved in computation. In the context of mobile phones, these layers include apps, middleware, mobile operating systems, app distribution systems, hardware, and cloud compute. Both Google and Apple tie developers to their operating systems by providing middleware that enables compatibility with a range of services – from mapping to payment software. Apple and Google are also crucial gatekeepers that use app stores to moderate what apps users can consume. Google and Facebook are the two largest and most influential providers of Software Development Kits (SDKs) that developers can integrate to measure app use. Big tech uses many of the stack layers act as control points to orchestrate the development, distribution, or governance of mobile software.

This creates several issues. For instance, big tech companies become de facto infrastructure providers for mobile services and functions such as navigation or mobile payment. They have established app stores as the main gateway to install apps on our phones, moderate the app offering, and determine what counts as a 'good' app worth distributing to users. They do not disclose tracking software embedded in app code and fuel data sharing and data leakages by

providing ad tech infrastructure. Importantly, users and developers carry the costs, as they are often locked into certain software even if they would like to use alternatives that align with their own values.

From app to cloud, software developers play a crucial role in providing technology alternatives that take into account different end-user groups and what they find important about mobile apps. In this report, we address developing parties that build different stack layers; such as small companies building apps, developer communities building software libraries, large cloud service providers, or companies developing new app distribution methods. Their end-user groups can be manifold and include citizens (including those with special needs), organizations that serve a public purpose (such as end-users in government institutions), organizations with a charitable mission and working in sensitive environments (such as NGOs in the social economy and humanitarian aid), or other mobile software developers who want to use more ethical software components that protect users.

In this report, we find that each of these groups find certain things important about mobile software. These findings are grounded in user values. We understand user values as convictions or beliefs that are important in someone's life. Values are not just preferences or opinions, but

describe why people judge something as important, worthy, or of quality.³

When interacting with mobile software, different things can be of value for users, for example: being connected to others (value of social connectedness), being able to separate professional from personal life (value of privacy), or being in control over the data one shares with others (value of data protection). Because these values can differ across groups, it is important for any developer working on a European and public mobile stack a) to understand what user groups they are developing for, b) what these find important about mobile software, and c) how the technology can be aligned with these values while avoiding harms.

This report provides developers and policy makers with a framework for how to understand, map out, and incorporate user values in mobile software development processes that aim to support user rights and interests. The report provides:

- A technological analysis of the mobile stack layers and their dependencies;
- An analytical framework for how user values can be related to stack layers, their technical design and the use cases they should promote;
- An illustrative list of user values that developers may consider when developing software; and
- An analysis of relevant EU-based legal frameworks for the development and evaluation of compliant mobile stack components when designing for user values.

This report is accompanied by a practical **Guide to Ethical Software Development**.⁴ The guide provides information on how developing parties can practically map out user values, choose and prioritise values, and translate them into design choices. It offers:

- Concrete methods for researching, documenting, and systematising user values. This includes methods to move from general

values to concrete descriptions of user actions and software designs.

- Methods for weighing values of different user groups against one another, curb possible biases when designing for one group and not another, and weigh the consequences of technology development to avoid value conflicts.

In sum, the report and its accompanying guide offer developers and policy makers with tools for aligning non-big tech mobile stack alternatives with the concrete needs of user groups, while taking into account EU law. Our documents help advance the development of public mobile stacks – stacks that are built with user groups around their values and needs, and regulated by public laws and the public values they represent.

1.1 Why aligning a mobile stack with user values?

There are several reasons for considering user values when building mobile stack layers:

- **Increased demand for value-aligned technology:**
There is an increased public awareness and demand for value-aligned technology. For instance, publicly funded organizations are increasingly expected to use software solutions that respect and protect the digital sovereignty of their organizations and countries. This is, for instance, reflected in policies that ban apps like TikTok from work telephones of civil servants, initiatives by cities and national governments to reduce dependencies from big tech and to protect themselves against loss of control over data flows, by adopting open source software. An increasing number of organizations seek to comply with European public values including transparency, lawfulness, accountability, fairness, collaboration, sustainability, privacy, digital sovereignty, and inclusiveness.

³ Friedman, B, and D. G. Hendry. (2019). *Value Sensitive Design: Shaping Technology with Moral Imagination*. MIT Press: Cambridge.

⁴ The guide is available at: <https://publicstack.net/guide-for-co-creation-of-ethical-software>

- **Public policy encourages the development of alternatives:** European public policy has targeted mobile software ecosystems as part of a wider effort to level the playing field between big tech, other market players, public institutions, and citizens. This includes European policies to create a Digital Single Market such as the Digital Markets Act, which
- **tackles, among other things, the separation of companies from app distribution, and messenger interoperability.** Values are also echoed in several EU-laws, including the EU Charter of Fundamental Rights, the General Data Protection Regulation, the Data Governance Act, and the Digital Services Act. In addition, there are policy debates on digital sovereignty and the development of Digital Public Infrastructure (DPI) (see section 1.2).
- **Developing promising alternatives to big tech that correspond to user's needs:** to develop alternative software that will be adopted by citizens, it is crucial that these technologies are in line with their values and needs.

In Europe and further afield efforts are already underway to develop digital stacks that are aligned with 'public values' and that provide digital public infrastructure. These are essential technological systems that are governed by the public, provide public goods, and advance the common good. Examples for initiatives include past funding programs such as the European Union's Next Generation Internet funding program, policy proposals such as the Eurostack, as well as concrete projects to build and improve free and open source mobile stack, such as the EU-funded Mobifree project, which this research is based on.

These efforts define policy priorities, steer investments into technology development, and organise concrete development activities. While

they play an important role for building non-big tech digital stack layers, they also leave questions unanswered as to how these stack layers should actually be built in line with 'public values'. In the coming section, we briefly present the push towards developing digital stack layers as digital public infrastructure, summarise its main arguments, and explain existing gaps according to critical commentators. Afterwards, we present a new framework, which explains why it is important to integrate user values and EU laws into the development and evaluation of digital stack layers, and how this can be accomplished.

1.2 Policy context: push towards value-aligned digital public infrastructure

Different mobile stack layers provide fundamental technologies on which public life runs. For example, apps for traffic routing, device authentication, and public services are deeply engrained in public life. The Covid-19 pandemic showed how big tech companies like Google and Apple can provide the technical foundation for epidemiological management: they provided contact tracing infrastructure and moderated permissible COVID-19-related apps through their app stores.⁵ Also digital identity apps mandated by government (see e.g. EU digital identity wallets) are built on top of existing mobile stack layers. They will likely provide a crucial identity layer to access digital government services. This means that mobile stack layers increasingly attain the status of infrastructure.

We define infrastructure as essential material basis to organise social life.⁶ An infrastructure is robust, widely shared among different user groups, widely accessible, routinely used, and essential.⁷ Some infrastructures attain the status of a public good, such as utilities or railroad networks.

⁵ Dieter, M., A. Helmond, N. Tkacz, F. van der Vlist, and E. Weltevrede. (2021). Pandemic Platform Governance: Mapping the Global Ecosystem of COVID-19 Response Apps. *Internet Policy Review* 10 (3). <https://doi.org/10.14763/2021.3.1568>

⁶ Larkin, B. (2013). The Politics and Poetics of Infrastructure. *Annual Review of Anthropology* 42 (1): 327–43. <https://doi.org/10.1146/annurev-anthro-092412-155522>

⁷ Edwards, P. N., Jackson, S. J., Bowker, G. C., and C. Knobel. (2007). Understanding infrastructure: dynamics, tensions, and design. Working paper, January. <http://deepblue.lib.umich.edu/handle/2027.42/49353>

Turning mobile software stacks into such 'digital public infrastructure' has become a cornerstone of European and global policy.

In Europe and beyond, digital public infrastructure is understood as a crucial means to advance different policy goals and address collective problems, including: digital independence and national technology sovereignty (see the Eurostack report), or economic competitiveness (see the Draghi report).⁸ The same applies for the European Union's legal frameworks we identified and analysed for this report. These regulate, for instance, fundamental rights protection, the EU's internal market ordering, data protection, and platform economies. These legal frameworks intend to protect European fundamental rights and freedoms of mobile ecosystem users (citizens, consumers). At the same time, they aim to support a level playing field for businesses, innovators, software developers and many others operating in the European Union's Internal Market.

Digital public infrastructure is not neutral. It embodies political agendas and values

Digital public infrastructure is never neutral; it embodies particular policy goals, assumes societal problems, and is optimised for particular values. Some researchers argue that on the international stage, 'digital public infrastructure' is a fuzzy term that can refer to metaphorical roads (digital infrastructure supporting economic development) as well as to media (digital infrastructure supporting democratic discourse).⁹ For these authors, these infrastructural metaphors each assume different societal problems and reflect different values. It is therefore crucial to ask what political agendas, values, problems and interests are shaping ideas and development activities around such infrastructures.

According to Keller and Krewer this also applies to European law-making.¹⁰ In a recent piece, they argue that Europe's policy frameworks centre specific societal issues, when regulating different parts of digital infrastructure. For example, social media is focused democratic discourse, whereas app stores are often framed around the value of competition. What is more, they point out that European policy pays more attention to certain parts of digital infrastructures than others (e.g. cloud infrastructure, as opposed to operating systems). For the authors, building a digital infrastructure aligned with public and democratic values requires a comprehensive understanding of the values across all components of a digital infrastructure. Otherwise, one part of the infrastructure could be optimised for one set of values, which can conflict with other parts of the infrastructure. As Keller and Krewer conclude, investments and developments of digital public infrastructure must clearly state the stack layers that are addressed, and make explicit what common problems are meant to be solved.

This analysis resonates with another influential and widely cited report by Eaves, Mazzucatto and Vasconcellos who argue that digital infrastructure must be built to align with public values, (such as openness and interoperability).¹¹ In addition, it must also advance publicly beneficial goals such as advancing (personal) autonomy, inclusion, and well-being, and it must be controlled by the public, for instance through rule-setting by democratically elected lawmakers, public development of infrastructure, or public funding. However, this report also acknowledges that it is unclear how public values that are embedded in the digital infrastructure (e.g. openness) can enable publicly beneficial goals (e. g. advancing

8 Bria, F., Timmers, P., and F. Gernone. (2025). EuroStack - A European Alternative for Digital Sovereignty. Bertelsmann Stiftung. Available at: <https://www.euro-stack.info/#report>

European Union. (2025). The Draghi Report on EU Competitiveness. Publications Office of the European Union. https://commission.europa.eu/topics/competitiveness/draghi-report_en

9 Rajendra-Nicolucci, C., and M. Samdub. (2024). What Is Digital Public Infrastructure? Towards More Specificity, Tech Policy Press. <https://www.techpolicy.press/what-is-digital-public-infrastructure-towards-more-specificity/>

10 Keller, P., and J. Krewer. 2024. Mapping the Debates About Strengthening Europe's Digital Infrastructure. <https://openfuture.eu/blog/mapping-the-debates-about-strengthening-europes-digital-infrastructure>

11 Eaves, D., Mazzucatto, M., and B Vasconcellos. (2024). Digital Public Infrastructure and Public Value: What is 'public' about DPI? Institute for Innovation and Public Purpose. <https://www.ucl.ac.uk/bartlett/publications/2024/mar/digital-public-infrastructure-and-public-value-what-public-about-dpi>

well-being). Moreover, the report assumes a stable set of 'public' values that shall guide the governance, design, and use of stack layers. As we will show, **our report presents an alternative perspective, and argues that these is not one, universal set of values, that corresponds to all user needs.**

Gaps in existing frameworks and research questions

These debates show that digital public infrastructure development is not a neutral process, but a matter of political choices. They also make clear that digital stacks can attain the status of digital public infrastructure when they are publicly controlled, when they provide essential services and are designed along public values and when they help advance the common good. However, existing frameworks leave a number of questions unanswered. Infrastructure is not only a set of basic services, but must also be appropriated by people to become infrastructure.¹² Socio-technical research into infrastructure reminds us that infrastructure can benefit some groups and harm others, for instance by providing access barriers or by excluding people.¹³ It is therefore important to ask how digital stack layers should be designed to fit not only general public values, but also different user groups. Optimising stack layers only for some people's values may cause harms for other groups, or be in conflict with EU laws. Furthermore, it is important to better understand how user values can be translated into design interventions for different stack layers, and how users actually use and experience them in their daily life. Lastly, regarding public rule-making, questions remain as to how stack layers can be made legally compliant with various EU laws.

To address these gaps, this report asks the following research questions:

- **What values do different end-user groups associate with different layers of a free and open source mobile stack built in Europe?**
- **What legal frameworks do mobile ecosystem developers have to take into account when designing and building a compliant mobile stack?**

In the following section, we present a framework to incorporate user values and EU laws into the development and evaluation of mobile stack layers. **We argue that a framework is needed that can facilitate iterative design and evaluation to account for alignment between stack layers, user values and relevant EU laws.**

1.3 Framework to align mobile stacks with user values and EU laws

User values and laws both need to be taken into consideration when building mobile stack layers. The famous claim that 'code is law'¹⁴ means that software formalises values by providing users with functions that steer their actions, or, on reverse, by constraining users in their choices. Values are selected by the software designers and developers. Software is therefore normative, and not neutral. In addition, software can be influenced by laws, for instance where it is designed in such a way that it, for instance, 'prohibits' hate speech. It is thus essential that developers are aware of how the design of a technology makes certain behaviours, choices and preferences more likely than others, or even steers these. What is more, the developing party is responsible for and must be able to demonstrate compliance with the law. We first introduce the mobile stack before explaining processes for value alignment and regulatory enforcement.

¹² Bowker, Geoffrey C., and Susan Leigh Star. (1999). *Sorting Things Out: Classification and Its Consequences*. MIT Press: Cambridge.

¹³ Star, Susan Leigh, and Karen Ruhleder. 1996. Steps Toward an Ecology of Infrastructure: Design and Access for Large Information Spaces. *Information Systems Research* 7 (1): 111–34. <https://doi.org/10.1287/isre.7.1.111>

¹⁴ See also <https://www.harvardmagazine.com/2000/01/code-is-law-html>

Stack layer	Examples for the mobile stack
Apps	Mobile applications are usually user-facing software that can be native (installed on a mobile OS), or web-based (run in a web browser). There is also a difference between system-level apps (apps that are pre-installed on an operating system) and regular apps. Apps are usually distributed via the app distribution layer (an app store), and more recently via direct download from the internet. Apps include middleware to be compatible with operating system functions.
App distribution (usually via app stores)	A system to make accessible, audit, and provide versioning services for mobile apps. It can include additional services, such as user screenings, app discovery, or payment service. App distribution can be connected to middleware. For instance, the middleware Google Play Integrity provides checks whether an app originates from the Google Play Store and has not tampered with. App distribution also has influence on the design of apps through app review processes. Examples: Proprietary app stores such as Google Play Store and Apple App Store, free and open-source distribution systems such as F-Droid or app package managers.
Middleware	Middleware is software acting between other software systems. Usually, this includes Software Development Kits (SDKs) and Application Programming Interfaces (APIs). Middleware is a crucial layer to provide additional functionality for apps (e. g. Firebase providing user analytics) but also a means to gather user data (e. g. Facebook Ads SDK) and to orchestrate ecosystems. Examples: Google Play Services, including libraries for authentication, advertising, security, and app permissions; MicroG (FOSS implementations of proprietary Google Play Services libraries); SDKs included as software libraries in apps
Operating system	A system that manages mobile hardware and resources and connects hardware and software. On mobile phones, Android Open Source Project (AOSP)-based operating systems and iOS are the most common operating systems, but Linux-based systems are also developed. Examples: iOS (Apple), Google-licensed operating systems, Android versions without license agreement with Google, such as /e/OS, or Lineage OS, Linux-based operating systems like Ubuntu Touch or postmarketOS.
Mobile phone hardware	Mobile phone hardware is the physical parts that make the phone work, including the processor (CPU), battery, display, cameras, speakers, microphones, and sensors - i.e., in the context of the mobile phone stack, it concerns the end user's terminal equipment. Examples: Mobile phone sensors, connectivity parts for mobile network, storage, circuit chips
Cloud hardware	Cloud servers are crucial to process app content and store user data. While in use, apps are communicating with servers to share user data. Our phones are therefore dependent on outsourced physical infrastructure to run. In addition, Apple and Google include cloud storage options like iCloud and Google Drive into mobile phones. Examples: Cloud servers connected to app functionality; Commercial (iCloud, Google Drive) and open source cloud services (Nextcloud, Murena Drive).

Table 1: Mobile stack layers

The mobile stack

Our mobile phones and apps are highly complex networked devices that are connected to many software and hardware components in order to function. These components are bundles of functionalities and services, often controlled by a few companies. Control over these components allows companies to coordinate and orchestrate the ecosystem of technologies making up our mobile phones. We call these components 'stack layers', drawing on the idea that computational systems are organised in different computational

layers. These layers can act like infrastructure if they provide the material infrastructure for other technologies to run on. We use the term 'stack layers' also because it is an increasingly common term in developer as well as policy circles. Table 1 provides an overview of different mobile stack layers. In this report, we focus on those stack layers that we tested with end-users in the Mobifree project. We are aware that the mobile stack can be unpacked in multiple ways. For example, a stack layer like the Android operating system can include other technical stacks than

the stacks we focus on.¹⁵ However, for the purposes of this report we focus on the stack layers that we have tested with users.

Top-level layers are more user-facing, while lower layers are more hidden from users and tend to be more infrastructural. This means that they provide foundation on which other layers can operate.¹⁶ In other words, mobile stack layers are not only isolated technologies, but act as systems that enforce rules (and values) onto other stack layers. This impacts both technologies and end-users. For example, Google's proprietary permission system for Android determines what kinds of data apps may request and whether data access is classified as sensitive or not.

To complicate this further, mobile stack layers, including apps, app stores, middleware, operating systems and cloud infrastructure, are highly interconnected and dependent from other computational infrastructures.¹⁷ By treating stack layers as networked, we emphasise that control over infrastructure often lies at the boundaries between computational systems and stack layers. Examples include computer interfaces like middleware (e. g. Google Play Services), or individual Application Programming Interfaces (APIs). On the one hand, these interfaces offer resources for third-parties to build on. However, they can also act as gatekeepers and control points because they can be used to impose rules on an entire ecosystem of software. This is often done by corporate owners, who use them to determine, for example, how software can talk to each other, or which software is interoperable or not.

Because the mobile stack is highly networked, efforts to build mobile stacks in alignment with user values and laws must not only focus on individual stack layers, but also consider the

structural dependencies between stack layers and how they can enable or hinder such efforts.

Treating stack layers as infrastructure also requires us to look at how different end-users use it in their everyday lives. Technology only becomes meaningful in use: only by using it in their daily lives, users are able to assess whether it is valuable to them. From this point of view, an infrastructure only becomes meaningful when it is being used and embedded in everyday use. This brings us to value alignment processes.

Framework for value alignment and legal compliance

Figure 1 shows how user values and laws relate to the development of the mobile stack and the roles that mobile developers (including user researchers and business developers) and policymakers can play in this process.

The right side of Figure 1 shows value alignment processes. The left side of Figure 1 shows regulatory enforcement and compliance processes. In the middle, we see different mobile stack layers and the entities and design processes to build them. Both, value alignment and regulatory enforcement have to be incorporated into the development and testing of mobile stack layers. In the following, we explain each component in more detail, starting with the mobile stack, then explaining how user values and EU laws relate to these layers.

It is important to note that our report does not provide a governance process for stack development. This governance process may assist developers to decide what user values should be considered, and how to balance these, if necessary, with other values, in alignment with EU law. The development of such governance processes is a task for future work and first advances have been made in the context of

15 Within each layer, one could identify additional stack layers, such as the Linux Kernel, Android Runtime, or Native Libraries for the Android OS layer, see https://www.researchgate.net/publication/280569434_A_Study_on_Smartphone_based_Operating_System

16 Larkin, B. (2013). "The Politics and Poetics of Infrastructure." *Annual Review of Anthropology* 42 (1): 327–43. <https://doi.org/10.1146/annurev-anthro-092412-155522>

17 Gerlitz, C., Helmond, A., Nieborg, D. B., and F. N. van der Vlist. (2019). Apps and Infrastructures – a Research Agenda. *Computational Culture*, 7. <http://computationalculture.net/apps-and-infrastructures-a-research-agenda/>

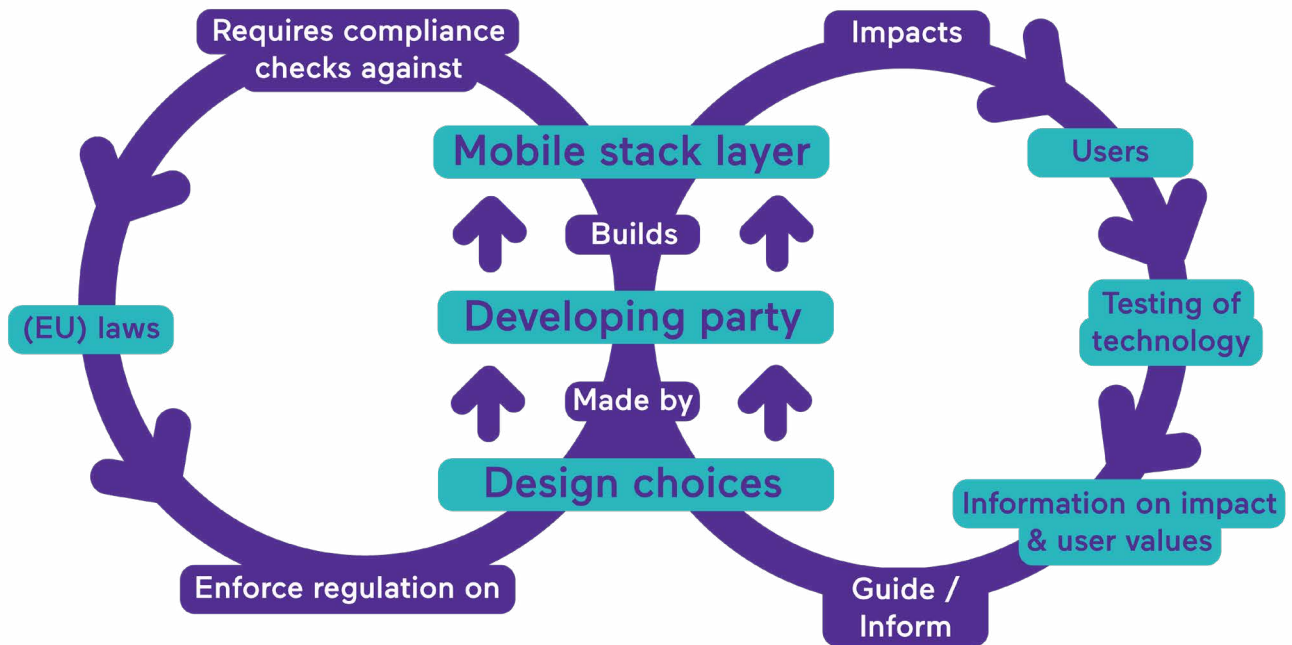


Figure 1: Framework for value alignment and legal compliance processes

decision-making models for Public AI development.¹⁸ We recommend that researchers and EU regulators, such as the European Data Protection Board (and other regulators) take up this task. Future frameworks can build on our work to develop governance processes, for instance by analysing the internal organisational structures and processes of mobile stack developers, or by testing the suitability of publicly accountable value-alignment and development processes.

Value alignment processes

The right side of Figure 1 shows value alignment processes. Because mobile stack layers impact users, they should be aligned with user values. We propose that value alignment processes consider at least two relationships between user values and stack layers: values manifest in how stack layers are built (its architecture, functions, expected use cases), and in how they are used (the enabled social interactions, user behaviour, interactions with other computational systems). We define

values broadly as the worth or quality that users assign to something. Users can value objects (e. g. specific mobile software and specific features), social interactions (e. g. how an app allows them to connect to others and manage personal relations) or entire states of the world (e. g. being certain that software does not violate privacy rights, that apps do not violate worker's rights, or that data is not used without one's consent).¹⁹

To give a concrete example: let's assume a user wants to search a useful new app on an app store. The app store may have values built into its architecture. For example, user ratings and downloads reflect the values of fame and popularity (how good is the app's esteem, how much attention does the app get). The user may choose an app with many reviews ('I use apps that many others are using') thereby acting in accordance with how the app store frames 'good' apps as 'popular' apps. Or the user may be distrustful of app reviews ('most reviews are fake') and rather choose an app whose technical

18 Ovadya, A., Redman, K., Thorburn, L., et al. (2025). "Position: Democratic AI Is Possible. The Democracy Levels Framework Shows How It Might Work." *Proceedings of the 42nd International Conference on Machine Learning*, October 6, 81930–61. Available at: <https://proceedings.mlr.press/v267/ovadya25a.html>

19 Heinrich, N. (2020). "A Pragmatic Redefinition of Value(s): Toward a General Model of Valuation." *Theory, Culture & Society* 37 (5): 75–94. <https://doi.org/10.1177/0263276420915993>

qualities fit the user’s interests.

User values are relevant for stack design, because they make clear that users can evaluate software as good or bad for many different reasons. User values are also helpful for developers because they become visible through concrete interactions with software and because they tell developers why people care about certain designs. This distinguishes user values from ‘preferences’ or ‘opinions’, which are often understood to be stable and which describe mere choices, without explaining why these choices are important for users.

There is a second reason why user values are relevant for stack design; users often prioritise what they find important and valuable in relation to their environment of use.²⁰ Software developers can therefore not always assume that certain software is useful, relevant, or valuable for users. Users may find themselves in social, organisational, and technological environments that influence whether they want or can use certain software. One can think of organisational rules or values that influence how employees organise their work and what features of software they find important to get the job done. If software developers ask why people value certain aspects of software (or not) they get a better understanding of how their software becomes relevant in concrete environments, and how it could be adjusted to fit these environments.

A value alignment process requires the documentation and analysis of user values, the translation of these values into social and technical design interventions, as well as the making, weighing, and evaluation of different design options. Below we describe a schematic overview for how we connected observed user values to design interventions and mobile stack layers, technologies, and concrete features. The outcome of value analysis is a list of user values, possible design norms, and concrete design

interventions that can be taken (in the table below, software developers could for instance implement a function to create private and professional accounts in a chat app, or develop a muting/invisibility function for professional chats after work hours).

User value	Design norm (based on user value)	Feature users pointed out	Stack layer; technology
Privacy (separation of private life from work life)	Being able to separate professional chats from personal ones after work.	Managing personal and professional chats	App layer; chat app

Table 2: Translation framework to link user values with design norms and features.

The alignment process can be integrated into existing agile and iterative software development phases. For instance, in planning phases, developing parties can run user tests, identify user values, and document them in possible design options. In the development phase, designers can weigh design choices, build and test prototypes with users. Finally, in the deployment phase, they can assess their use. More information on how this value alignment process can be organised is available in our Guide to Ethical Software Development.

Regulatory enforcement processes

The left side of Figure 1 shows regulatory enforcement processes. Laws are created by democratically elected lawmakers. These laws are enforced by public authorities (regulators such as supervisory authorities, or courts) to ensure that the law is complied with by citizens, consumers, public entities or companies. This includes mobile software developers that build the mobile stack. Entities developing mobile stack layers need to take into account laws that apply to them. This ensures that their software and hardware produce (user) behaviours that are compliant with what democratic regulators deem acceptable and that

²⁰ Le Dantec, C. A., Shehan Poole, E., and Susan P. Wyche. (2009). Values as Lived Experience: Evolving Value Sensitive Design in Support of Value Discovery. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, April 4, 1141–50. Available at: <https://doi.org/10.1145/1518701.1518875>

respect users' fundamental rights and freedoms. Notably, most laws under review explicitly recognise the rights and business interests of software developers as entrepreneurs contributing to fostering innovation in the EU's internal market.²¹

It is important for developers of mobile stack layers to consider both user values and laws. This is because there is a notable interplay between values and laws. Some laws clearly define what desired behaviour is, and can be easily applied to a specific situation (for instance, you are not allowed to steal someone's car). Yet laws that apply in the digital society are often more ambiguous. Many aspects of laws addressing behaviour in the digital society need interpretation. Such interpretation is sometimes proved for by a court, or by a regulator (such as the European Data Protection Board (EDPB), or by the Recitals preceding the text of the specific EU-law that explain what the legislator has meant with a specific provision).

Regulatory guidance aims to increase legal certainty among developers about the interpretation of the laws; and guidance may be expected as regards the more recently adopted laws. **Stakeholders and others interested in contributing to such guidance are usually invited by the regulator to respond to a concept version which is published on the regulator's website. Processes for regulatory guidance bringing some legal clarity may take time, whereas developer activities in the mobile stack are overall dynamic.**

Summary: principles to relate user values, laws and mobile stack layers

1. How the mobile stack is built influences how people experience it in daily life. Firstly, values are built into the material characteristics and functions of mobile stacks. These characteristics enable and constrain how humans and technologies can behave, and they can embody social and cultural values. The material features of mobile stack layers can be directly experienced and tested with users (e.g. testing certain features in the graphical user interface), but they can also be invisible or not entirely understandable to users and harm them without users necessarily noticing it. Developers therefore have to consider in what user facing or invisible technologies values have to be implemented, and how they inform users.

2. Users experience values in relation to their daily life. These experienced values can differ from abstract public values. As we will show in the next chapter, multiple values come to play in concrete situations. Rather than deciding on a set of abstract 'public values', we argue that what is valuable must be derived from user's concrete experiences. Different user groups may ascribe different values to technologies, or may even interpret the same value in different ways. Which values they choose, depends on the social context in which they use the technology. To make a value meaningful in practice, it is essential to look at what users concretely need (e.g. in terms of functionalities) to be able to act upon this value. For example, to act in accordance with the value of 'privacy', users may need specific functionalities and explanations to feel that they can make informed decisions about their privacy.

²¹ Laws including the General Data Protection Regulation the Digital Governance Act, the Data Act, the AI Act, the Cyber Resilience Act or the Digital Markets Act all refer to the law's functioning in the context of the EU internal market (Janssen et al., Towards compliant mobile Ecosystems Design. Mapping and exploring EU Law applying to mobile stack developers, see sections 5.1, 5.4, 5.5, 5.6, 5.6, 5.7 available at: <https://zenodo.org/records/22918307>).

3. Values can differ across end-users and may change over time, or depending on the context in which they apply. A continuous, dynamic and iterative process is needed for value alignment.

Value-alignment is not a one-off process. The mobile stack consists of infrastructures, used by different end-user groups with different needs and understandings of what is valuable. As infrastructure, some mobile stack layers can also benefit some users, but harm others. What people perceive as important and valuable may also change over time and is contextual. User groups may experience software differently because they use specific sets of technologies and use specific features for different purposes. Designing a value-aligned mobile stack is about creating some standard functionality that steers user actions in beneficial ways, while leaving space for different user groups to act according to what they find important. To accomplish this, iterative value-alignment processes are needed that document values, translate them into functions and user actions, evaluates the process, and deals with value tensions in the stack.

4. Experienced values require translation into different mobile stack layers. Once documented, user values have to be put in format that makes it possible for mobile software developers to use them for their software development. Here, it is important to remember that software is normative: how it is designed by developers directly impacts what actions are possible for users. To incorporate user values, developers can translate these values into design requirements. For instance, the value of privacy could be defined as 'being able to determine who can contact a user'. Such a statement is very close to the definition of a user story and similar formats used in software development, while also providing providing an actionable description of how a user understands a value like privacy.

5. User values can help software developers to align with the intention of the legislator. Developer parties can refer to user values to shape the definition and interpretation of open-natured laws to design mobile stack layers in accordance with laws and the public values they seek to promote. For example, the way privacy is described in the law may not provide enough guidance for developers, so a developer could look at user values in order to find an appropriate approach to design for privacy. Consulting values, however, may entail uncertainties for developers, as values themselves often reflect open norms. We therefore urge and recommend policymakers to develop practical and guidelines for developers, in particular for those wishing to design compliant software.

2 Our methods

2.1 How we studied user values

This report is based on the research conducted in **Mobifree**, a EU-funded project that aims to develop a mobile software ecosystem that puts citizens' values at its centre. The software we tested during this project aims to be 'ethical tech', which respects people and planet. One of the main goals was to offer an alternative to big tech mobile stacks and their ecosystems, which is in line with the values of its users. In this report, we concentrate on the value-based user research that we conducted, as well as the legal analysis concerning relevant legislation for mobile software developers.

Our empirical research followed a value-sensitive design approach to identify user values and relevant design interventions following from these values. Value-sensitive design aims to build technologies in line with the values of users, by designing technologies with positive impact on different user groups while avoiding harm for some groups. We involved four user groups for whom a value-aligned mobile stack may be helpful, and who can experience different harms: civil servants, mobile app developers, humanitarian aid providers and young adults. To test the software in practice, we organised workshops to understand their existing use cases, problems they encounter with mobile phones, and ideas for an ideal mobile phone (n=102). From this we derived a first set of user values and topics to explore in four pilot tests (one per user group). For the pilots, participants (n=127) tested over eight weeks different Mobifree software, developed by partners in the project. Testers received 'test instructions' every week and were asked to complete a short online survey at the end of each week. At the end of each pilot, we organised a focus group with a sample of five to seven participants to discuss recurring themes in the survey responses.

In our analysis, we followed VSD's goal to identify different user values and possible value misalignments and tensions between these groups that design should address.²² To operationalise this approach, we developed a data analysis framework (see a shortened version in section 1.3) that made it possible to systematically connect participant responses to specific aspects of the technologies. For each instance in which a value emerged, we documented: (1) the technology, (2) the specific function of feature users talked about, (3) what people think about how the feature is designed or how it should be used, (4) the value that people care about in this instance, (5) the context and empirical evidence in which people provided feedback. To make user feedback and values actionable, we phrased them as descriptions of desired user actions (for instance, for privacy in the context of chat apps, we identified that 'being able to keep one's private life separate from work life' is an user action that should be translated into a feature (e.g. multiple account management)).

The following technologies were tested in the four pilots:

- /e/OS: a 'deGoogled' operating system. It comes with its own pre-installed apps, including the App Lounge, which we also tested. /e/OS has its own privacy features, which make it possible to block app trackers, hide one's IP address and use a fake geolocation.
- App Lounge: /e/OS's own app store. It sources both FOSS and proprietary apps. It gives users a choice between logging in anonymously or using a Google account.
- F-Droid: an alternative app store that only distributes FOSS apps. Different user groups tested different parts of F-Droid. All groups tested the F-Droid app for downloading FOSS

²² Friedman, B, and D. G. Hendry. (2019). *Value Sensitive Design: Shaping Technology with Moral Imagination*. MIT Press: Cambridge.

apps. The developers also tested the F-Droid app distribution system, including the tools Fastlane and the DAPPER registry.

- Delta Chat: an alternative messaging app. It is privacy-focused and decentralised.
- Conversations: an alternative messaging app. Privacy-focused, federated and built on the XMPP protocol.
- Murena Workspace: cloud collaboration software, including documents, spreadsheets and calendar.
- Murena’s Mobile Device Management (MDM): MDM software for managing a fleet of work devices. This was only tested by the civil servant user group.
- MicroG: an alternative for Google Play Services for Android systems.
- Humanitarian software: this software was specifically developed for the humanitarian user group. Commit Global developed PTSD Help (to help humanitarian workers recenter in stressful and traumatic situations), OneRoof (to manage emergency shelters), VIC and RVM (to manage resources and volunteers).
- Rapid.Space developed a portable communications system to provide connectivity for humanitarian professionals in emergency contexts that have no existing infrastructure.

Readers interested in how we applied the method to ideate design interventions can consult our **Guide to Ethical Software Development**. This guide helps developers to reflect on salient stakeholders of their technology, elicit values from users and, and to systematize and prioritize values and design requirements that follow from the values.

2.2 How we mapped applicable laws

Selected laws

For this report, we selected fourteen EU laws that apply in different manners to activities of mobile ecosystem developers, including the developers of the Mobifree consortium.

1. EU Charter of Fundamental Rights	CFR
2. General Data Protection Regulation	GDPR
3. e-Privacy Directive	e-Privacy Directive
4. Free Flow of Non-personal Data Directive	FFNPD
5. Data Governance Act	DGA
6. Data Act	Data Act
7. AI-Act	AIA
8. Product Liability Directive	PLD
9. Cyber Resilience Act	CRA
10. Digital Markets Act	DSA
11. Digital Services Act	DMA
12. Corporate Sustainability Reporting Directive	CSRD
13. Corporate Sustainability Due Diligence Directive	CSDDD
14. Equal Access Act	EAA

Why these laws

Each of these laws stipulate different legal norms and regulate different types of desired behaviours, and aim to enforce legal compliance with these norms.²³ The selected EU laws apply to mobile ecosystems developers operating within the EU. The EU aims to create a level playing field for competition, with laws equally applying to all mobile ecosystem developers. Being aware of that comparable laws are under development in other legal systems outside of the EU context (such as Canada, India or California), it would complicate our legal research in terms of the size of this report. We therefore restrict ourselves to developers offering their services and goods on EU territory, or goods and services to EU citizens.

²³ “A legal norm is a binding rule, that organisations of sovereign power promulgate and enforce in order to regulate social relations. Legal norms determine the rights and duties of individuals who are the subjects of legal relations within the governing jurisdiction at a given point in time”, Krasner, S. D. (1982) Structural Causes and Regime Consequences: Regimes as Intervening Variables *International Organization* 36(2), 185–205, <http://www.jstor.org/stable/2706520>.

Analysis of how the laws apply to software developers

In our report, we explain how the selected laws apply in some form to most app, app store, cloud and operating systems developers, but each law applies to different aspects of developers and their goods and services offered.²⁴ Some laws apply only to some very large sized businesses (Digital Markets Act)²⁵, while legal aspects of other laws focus on small and medium-sized enterprises (Digital Services Act).²⁶ Some laws specifically focus on user rights to their data (GDPR),²⁷ while others regulate a developer's obligations relating to cyber resilience (Cyber Resilience Act)²⁸ or a developer's reporting obligations (Corporate Sustainability Due Diligence Directive)²⁹. We map and explore for each law the relevant Articles and Article sections that may apply operating systems developers, cloud developers, app store developers and app developers; we map the underlying values of these laws, and explore, at a generic level, what mobile ecosystem developers, including partners from Mobifree, should do to be compliant with the applicable laws that apply to them. We unpack these in section 5 of this report. The underlying **report** and its **methodology** are stored in the publicly accessible repository Zenodo.³⁰

We also carried out semi-structured interviews in the field. Interview partners were working in the field, mostly as a technical developer or as a manager in an organisation which is part of the Mobifree Consortium. We aimed at getting better acquainted with (i) their level of knowledge and understanding potential applicable laws that might impact the design of their app, app store, cloud service or their software design, (ii) values they prioritise to protect and (iii) where uncertainties over legal compliance exist (Section 5 of the report).

For a quick overview of, respectively, the values, the applicability of the law to stack developers, the purpose of the law, its scope, its key definitions, its addressees and compliance aspects, we added a **Table briefly describing laws on all these aspects** in the repository as well in Zenodo. The reports of the interviews contain personal data and are therefore not publicly accessible.³¹

24 Janssen et al (2026), Section 1.1 available at: <https://zenodo.org/records/22918307>

25 Janssen et al (2026), Section 5.9 available at: <https://zenodo.org/records/22918307>

26 Janssen et al (2026), Section 6.4 available at: <https://zenodo.org/records/22918307>

27 Janssen et al (2026), Section 6.1 available at: <https://zenodo.org/records/22918307>

28 Janssen et al (2026), Section 6.7 available at: <https://zenodo.org/records/22918307>

29 Janssen et al (2026) Section 6.13 available at: <https://zenodo.org/records/22918307>

30 Janssen et al (2026) available at: <https://zenodo.org/records/22918307>

31 For more information, please contact h.l.janssen@uva.nl

3 User values and the mobile stack

In this section, we present our main empirical findings. We found that:

- **Values are not interpreted uniformly:** different user groups prioritise different values, and can interpret the same values in their own way.
- **There is a gap between values as abstract concepts and how users perceive them in practice:** translating values into software in a way that corresponds to users' needs requires involving users in the design process.
- **There are dependencies between the different layers of the stack:** values that appear to concern one technology, may in fact need to be addressed at another layer of the stack.

For each finding, we provide illustrative examples from our research, referring to different Mobifree technologies and stack layers. A comprehensive overview of all user values we identified for different stack layers can be found in our open research data, available at Zenodo.³²

3.1 User groups prioritise different values

In policy discussions on embedding 'public values' in technology, there often is an implicit assumption that public values are universal. While abstract public values can be a good start, it is essential to see them in relation to context of use. We found that what users value is directly related to the context in which they use their phone.

This meant that the four user groups - young adults, civil servants, mobile app developers and humanitarian aid providers - noticed different things about the software they tested. For instance, as the young adults used the phone in their personal lives, they looked at it through a lifestyle lens. They valued customisability, privacy

and simplicity. The civil servants, on the other hand, reviewed the phone in relation to their context of work. Privacy, data protection and the trustworthiness of the software were particularly important to this group. The mobile app developers assessed the software in relation to their professional context as well. This meant that this group was uniquely interested in the more technical aspects of the software, and they highlighted the values of security and provenance, particularly in relation to the app distribution process. Finally, the humanitarian aid providers also specifically focused on their work context. As their profession involves working with sensitive information and vulnerable groups, they emphasised privacy and security to protect their own data as well as that of beneficiaries.

In the table below, we look at the differences and similarities in values more closely. Not only do different user groups emphasise different values; when they do pick the same value they conceptualise it in different ways. We demonstrate this by looking at one stack layer: app distribution, using the App Lounge as an example. The App Lounge is an alternative to the two main app stores, Google Play and the Apple App Store, and comes as a pre-installed app with the /e/OS operating system. It offers both free and open source and proprietary apps, and makes it possible for users to log in anonymously.

Not only do different user groups bring up different values associated with the App Lounge. If users share values such as security, privacy or understandability, they interpret these values differently, and in accordance with their social and organisational context. This demonstrates why abstract public values alone provide an insufficient basis for technology design. Listening to different

³² Open research data on user values can be found at: <https://zenodo.org/records/22918474>

Value	Young adults	Civil servants	Mobile app developers	Humanitarian aid providers
Findability	Relate findability to discovery; want to be able to discover free and open source alternatives; point out that they miss recommendations based on searches and ratings and reviews, as these features help them decide which apps to download.	Relate findability to work-related access: were able to find their regularly used apps, but organisational policy limits their ability to use free and open source alternatives.	-	Relate findability to work-related access. Would prefer to be able to filter according to their own priorities and not just free and open source alternatives.
Understandability	Relate understandability to usability; need clear information and an intuitive interface to be able to use App Lounge.	Relate understandability to coherence and familiarity; need recommendations that match the values of the app lounge (so no mainstream commercial apps); need explanations of unfamiliar app formats such as apks to feel safe enough to download them.	Relate understandability to being able to understand the origin of problems with software so that it is possible to troubleshoot them.	Relate understandability to informed decision-making; need more information about privacy, security and app-vetting processes before they are able to trust the app.
Reliability	Relate reliability to the performance of the software; the software should perform in the same way when used repeatedly.	-	Relate reliability to performance of the software: errors, slow loading and unreliable anonymous mode undermine confidence in the App Lounge.	-
Safety and security	-	Relate safety and security to familiarity: unfamiliar downloads (such as apks) felt unsafe.	Relate security to being informed about security measures; need to be able to rely that an app distribution system provides security checks and quality controls when it gathers app data from an upstream app distribution system.	Relate security to being informed about security precautions and app vetting processes as they work in a sensitive environment where security is paramount.
Provenance and trust	Relate provenance/trust to being able to see what the origin of the apps is and what sources they are imported from.	-	Relate provenance/trust to being able to see the origin of the apps. Most developers generally understood which sources apps are imported from, but did not understand why there they were able to find different versions of the same app in the App Lounge – and where those come from.	-
Choice	Related choice to the ability to explore different types of apps. Were positive about the fact that both mainstream and open source apps were available in the App Lounge.	Related choice to the ability to explore different types of apps. Also positive about the availability of mainstream and open source apps, but expressed confusion at the recommended apps.	-	-

Table 3: Example user values related to App Lounge

user groups makes visible how values are interpreted in specific social and organisational contexts, and what users need for them to work in practice. This challenges a one-size-fits-all approach to public values in technology policy: rather than treating (public) values as universal requirements, policy and design need to account for how different user groups experience and prioritise them.

3.2 The gap between values and practice

In the previous section, we showed that different user groups emphasise and interpret values in different ways. Our findings also reveal an underlying issue: even when values are incorporated into a technology, it does not guarantee that users can understand or act on it. This is a finding we observed across several technologies, including /e/OS privacy features, F-Droid, and chat apps Delta Chat and Conversations. The users we consulted were often unable to understand features, or make informed decisions based on them. In other words, there is a gap between how values are currently implemented in the technology, and what is needed for users to meaningfully act on these values in practice.

In general, all four user groups were very positive about the concept of FOSS: they were motivated to find alternatives to software developed by big tech companies. However, our testers pointed out that the software was not always easy to understand, and mentioned that some applications seemed to be aimed at users who have an above-average knowledge of technology. For example, testers in all four user groups commented on the understandability of technical terms in several of the Mobifree technologies. This becomes most visible in relation to the values of privacy and security. For example, /e/OS, the 'degoogled' operating system, comes with special privacy features: users can block app trackers, hide their IP address, or use a fake geolocation. While these features were received very positively, understandability issues occurred in several user groups. For example, one civil servant remarked that when they tried to hide their IP address, they

received a notification that that this might result in them being permanently banned from their email account. As they remarked:

'That's not something I want to risk and because I don't have specific knowledge about technical stuff of phones I'm afraid to use this setting. If this setting could be made dummy proof so people like me don't have to worry about it (so the setting will chose not to be applied for those risky things that could get you banned) I would happily use it.'

In other words, without sufficient knowledge to understand the consequences of enabling the setting, the tester did not feel able to use it. This is important, because it directly impacts their ability to make informed decisions about their privacy. Similar issues came up in relation to other technologies. For example, one of the humanitarian testers remarked in relation to the 'anti-features' on F-Droid app store, which warn users about potential features that clash with open source values, such as ads or app tracking:

'I think the idea is amazing, it just doesn't really explain it well to non-technical users. For example: "this app depends entirely on a certain instance of a network service" -> Rely on hardcoded Nostr relays. What ??? I consider myself to be relatively technical, but even this doesn't really clarify what the actual impact would be on me.'

Also the privacy settings of the chat apps, Conversations and Delta Chat, were not always easy to understand. As one of the young adult testers remarked about Conversations:

'There is lots of technical terms, OEM encryption option for PGP, which seems to be good options, but I wouldn't know what to select. The app seems to be made for people that already have knowledge on this topic and could benefit from some small explanations.'

They key point that unites these different examples, is that these features are created to protect users' privacy and security. However, when users cannot understand how privacy features work, or what the consequences of

enabling them are, they cannot make fully informed decisions about their privacy. Importantly, our research therefore reveals a gap between how 'privacy' is embedded in the software, and how users experience it in practice. This highlights the need to move beyond abstract values and examine in an iterative fashion how they are understood in practice, and what users require for them to work in the context in which they use the software.

3.3 Governing stack layers

We have also observed that one value-related issue that users encounter in one part of the stack, may have its cause (or its solution) in another stack layer. Therefore, it is important to consider values not only in relation to individual technologies, but across the (interconnected) layers of the stack. Developing value-aligned stack layers in practice is not only a question of designing individual technologies, but also to understand how they interact with the wider stack.

One example revolves around the trustworthiness of software. Testers across user groups described some applications as 'unpolished' or not trustworthy, which impacted how trustworthy they perceived them to be. This came up, for instance, in relation to the F-Droid app store. One of the civil servants remarked, for example, that 'the store has a real cowboy vibe, anything can be on there.' This was not a point that only came up in the non-technical user groups; also some developer testers mentioned that their trust was impacted:

'But also all the random apps in the F-Droid store that have simple icons made by some icon generator seems way less trustworthy. [...] when I see an app or OS that has put a lot of effort into making things work well (UX) or look good (UI) then I am more likely to trust they put the same amount of effort into making the app secure and work well on the backend.'

Therefore, aesthetics matter: a perceived lack of polish or smoothness directly impacts the trust that users have in an application. On the surface, app quality may appear to be a UX issue. However, it is also shaped by app-store policies governing which apps can be included and under what conditions. App stores can impose criteria that apps need to fulfil in order to be published in the store, which can include requirement on looks and UX. Most users' perceptions of quality and safety are likely shaped by the Google Play or the Apple App Store. Both are documented to influence the app design through app review processes. F-Droid, by contrast, deliberately does not include user interface design as a criterion for app review to prioritise developer autonomy. This is a deliberate choice, and deciding to create more rules around the design of apps, would require a shift on the side of F-Droid. This means that there is a tension between user and developer values. More fundamentally, however, this example illustrates that the layers of the stack are interconnected: the way a value is experienced at one layer may depend on decisions and conditions at others.

Dependencies across the stack

The second example relates to the App Lounge's 'anonymous mode', which allows users to log into the App Lounge without logging in. Testers were very positive about not having to sign in, as they felt their privacy was better protected. At the same time, they pointed out that this option did not always work, which meant that they had to sign in using a Google account. Testers perceived this as a striking and sometimes confusing option in a deGoogled ecosystem.

However, while this seems like a (problematic) dependency on Google on the interface level, the dependency runs deeper. According to /e/OS's documentation³³, App Lounge accesses the Google Play catalogue through Google's API. This dependency arises because App Lounge provides access to both FOSS and mainstream proprietary Android apps, such as WhatsApp and banking apps. Therefore, in order to be able to offer certain popular apps, the App Lounge is dependent on using a Google service.

This illustrates the difficulty of creating a fully deGoogled ecosystem: even when Google is removed from the user-facing interface, its infrastructure may remain embedded in the underlying technologies on which the ecosystem depends. This has broader implications for developing ethical software: only looking at the visible layer can obscure underlying dependencies that continue to shape the infrastructure.

In sum, the empirical examples above illustrate the importance of incorporating users in ethical software design: in order to develop software that allows different user groups to act on values in a meaningful way, the software needs to be attuned to their interpretation of values and their needs. As such, policy and software development needs to go beyond abstract 'public values' and define values in practice, together with users. What is more, our research highlights the importance of mapping dependencies across the stack: only through a full-stack view on software, dependencies on commercial (big tech) parties can be reduced.

³³ /e/OS Community. "App Lounge - Know All About It," June 30, 2022. <https://community.e.foundation/t/app-lounge-know-all-about-it/42331>

4 EU Law applying to user values & the mobile stack

Report on EU Law applying to user values & the mobile stack

We mapped and analysed the legal frameworks that may apply to mobile ecosystem developers, which also include Mobifree developers. We identified the following laws:

1. EU Charter of Fundamental Rights	CFR
2. General Data Protection Regulation	GDPR
3. e-Privacy Directive	e-Privacy Directive
4. Free Flow of Non-Personal Data Directive	FFNPD
5. Data Act	Data Act
6. AI-Act	AIA
7. Product Liability Directive	PLD
8. Data Governance Act	DGA
9. Cyber Resilience Act	CRA
10. Digital Services Act	DSA
11. Digital Markets Act	DMA
12. Equal Access Act	EAA
13. Corporate Sustainability Reporting Directive	CSRD
14. Corporate Sustainability Due Diligence Directive	CSDDD

Overview: table of laws & key points to consider by mobile stack developers

We approached and described the laws in a structured way to highlight parts that likely apply to (some of) the mobile ecosystems developers. We briefly present these descriptions and analyses in our table summarising each law in Zenodo. The details are described and analysed in our more extensive report.³⁴

We now evaluate our findings from the legal findings and analyses of applicability to user rights and to mobile ecosystems developers obligations, and relate them where possible to the

key findings from the values related analyses undertaken by Waag Futurelab.

The role of law in the mobile stack
In the context of mobile ecosystem developers, EU law sets limitations on what a mobile ecosystem developer do with a user's (personal) data (General Data Protection Regulation)³⁵, while prescribing methods for responsible, safe and secure processing of that (or other) data (e.g., GDPR, Cyber resilience Act)³⁶. EU law not only protects user data, it also guards small sized, medium sized and micro-sized developers interests as well as non-commercial Free and Open software (FOSS) developers against large, powerful companies, also referred to as 'gatekeepers' (e.g., Digital Markets Act³⁷, Digital Services Act,³⁸ Cyber Resilience Act³⁹, Product Liability Directive⁴⁰).

Observations of law in practice

From the semi-structured interviews, we noted that open-source and privacy-focused digital projects face a complex and rapidly evolving EU regulatory environment. Yet many of the developers interviewed lack legal expertise, resources, and practical guidance needed to navigate it. GDPR is the most commonly recognised framework, while awareness and experience with newer legislation such as the DMA, DSA, Data Act, Cyber Resilience Act (CRA), AI Act, Accessibility Act, or Product Liability Directive vary considerably. The main problem is not necessarily a lack of commitment to privacy, security or responsible development; rather, developers may struggle

34 Janssen et al. (2026) . available at: <https://zenodo.org/records/22918307>

35 Janssen et al (2026), Section 6.1 available at: <https://zenodo.org/records/22918307>

36 Janssen et al (2026), Section 6.7 available at: <https://zenodo.org/records/22918307>

37 Janssen et al (2026), Section 6.9 available at: <https://zenodo.org/records/22918307>

38 Janssen et al (2026), Section 6.10 available at: <https://zenodo.org/records/22918307>

39 Janssen et al (2026), Section 6.7 available at: <https://zenodo.org/records/22918307>

40 Janssen et al (2026), Section 6.8 available at: <https://zenodo.org/records/22918307>

Reported user value	Legal value	Law
Human centric approach	Respect for human rights	CFR, GDPR, AI act, EAA
Privacy	Personal autonomy Privacy Data protection Communication secrecy Human dignity	CFR, GDPR CFR, GDPR CFR, GDPR GDPR ePrivacy Directive CFR, AI Act
Ease of use	-	-
Understandability	Transparency towards users	GDPR, ePrivacy
Reliability	Accountability, transparency	GDPR, Data Act, DSA
Safety & security	Security	GDPR, Data Act, AI Act, CRA, DSA
Provenance, trust	Documentation obligations, user access to data & data processing	GDPR
Choice	Personal autonomy User control Avoid user lock-in	GDPR GDPR, ePrivacy Data Act
	Data sovereignty & data autonomy	DGA, Data Act
	Liability	PLD
	Fair & contestable markets (protect micro and SMEs)	DMA
	Investor access to assets information	CSDR
	Sustainable & responsible corporate behaviour	CSDDD

Table 4: Reported user values and related legal values in the laws applying to mobile ecosystems

to translate their technical and ethical practices into a clear understanding of their legal obligations. The strongest recurring need is for practical, accessible, real-time and plain regulatory guidance connecting technical characteristics of a project to the applicable legislation. Such guidance should particularly account for the realities of micro-sized or small teams, open-source development, decentralised infrastructure, commercial/non-commercial hybrids, and rapidly changing EU digital regulation.

User values and EU law

From the empirical research, several values users deem important were expressed by the four types of users involved. These values can be found in Table 4 in the left column. While the weight attached to a specific value varied across the populations and contexts, these values are also acknowledged across several different EU laws (see for the specific values, Table 4, second column). We also listed the specific laws where the values can be found in Table 4, third column.

Observations

Notably, nearly all values users deem important from the empirical research are reflected in EU laws. Table 4 often mentions the GDPR; this can be explained against the background that the GDPR protects a user's personal data. All mobile stack developers may process personal data; whenever they process a user's personal data, the GDPR applies. Other than the GDPR, the ePrivacy Directive, the AI Act, the Data Act, the Digital Services Act and other laws may also apply - all depending on the specific purpose for and the specific context in which the developer processes (personal) user data. Other than user values, legal obligations are not an option to consider. Laws must simply be complied with. Laws reflect enforceable legal norms that are to be distinguished from values or norms. That is, mobile ecosystems developers must take user rights as well as the corresponding (and other) legal obligations into account. Given laws involved largely reflect user values, a large part of these might already be covered and protected by the respective laws.

The values transparency and accountability – an example

Regarding the protection of user values, we briefly explain one (GDPR-based) example to clarify how user values are dealt with and what developers should do to effectuate their protection.

The GDPR obliges developers to operate according to the values of transparency and accountability. Before starting the processing any personal data of users, developers must inform the user in a concise, transparent form using clear and plain language about, amongst other, all type(s) of data they seek to process, the purposes of the processing, the recipients of the data and the storage terms applied to the data. Users must, based on that information, consent with the developer's intended data processing, prior to the processing itself.

The value of user control – an example

EU law aims to give users some form of **control over** their data as well.⁴¹ Users can exercise their right against a developer to get access to the data that is processed about them; they can ask a developer to erase their data or port their data to another developer at their request (GDPR).⁴² By accessing their data, a user that distrusts the developer's processing can also determine to withdraw consent (meaning that the developer must discard all personal data, including personal data already processed).⁴³

Ease of use as a user value?

The value 'ease of use' was not represented in any of the laws. In our report however, we described this value as a desire for 'seamless' applications.⁴⁴

As seamlessness is important for users, developers may push their limits to offer users seamless systems. Yet, with seamlessness, risks to the transparency of a system's processing may occur: systems without seams often operate in less or non-transparent ways. That is, seams

represent a value as well: a transparency seam 'forces' users to take note of the information about the developer's processing, intentions and methods, and to 'consciously' give their consent. The European Data Protection Board identified several examples where seamlessness makes it more challenging in mobile ecosystems contexts to adequately inform users about the processing of their personal data, as compared to processing that data in the desktop environment.⁴⁵

In other words, developers may offer systems with less seams, but within the limits of the law (the GDPR's transparency requirements must be complied with).

Developer compliance with user values through legal obligations

To effectively realise user (and other) rights, laws usually puts legal obligations on developers processing data for their own purposes; in the context of this study, we focus on developers in the mobile stack. These obligations can be found across several EU laws, such as the GDPR, AI-Act (AIA), or the Data Act (DA). In our report, we explored what laws apply to developers across the stack, and where possible, we distinguished at what stack layer (app, app store, cloud service or operating system) laws might apply.

41 Janssen et al (2026), Section 6.1 available at: <https://zenodo.org/records/22918307>

42 Janssen et al (2026), Section 6.1 available at: <https://zenodo.org/records/22918307>

43 Article 7 GDPR.

44 Janssen et al. (2026) Section 3.3. available at: <https://zenodo.org/records/22918307>

45 European Data Protection Supervisory Authority (2016) Guidelines on the protection of personal data processed by mobile applications provided by European Union institutions (2016).

Complex laws and growing numbers of stack developers: guidance desired

It is for developers themselves to explore and assume a law's obligations pertaining to their users' rights. Given the increasing number of developers across the mobile stack and the law's complexities and possible interactions with other laws, we urge EU regulators to assist mobile stack developers with their obligations by developing guidance, ideally in cooperation with stack developers.⁴⁶

Mobile stack developers that are new to the internal market and to the legal obligations they may have to comply with - here referred to as 'tech startup developers' - would particularly be served with tailored guidance. Such startups, who are often innovating at the cutting-edge - typically push the boundaries of technologies that typically lack established legal and data protection best-practices.⁴⁷

Initial decisions taken by startups could, however, well have long-term impacts, and their actions may inform (for better or for worse) how their technologies are implemented, deployed and perceived for years to come.⁴⁸

Ensuring that the innovations and practices of those developers are sound, appropriate and acceptable warrants regulatory attention, and should therefore be a high priority for regulators.

Protecting a developers' interests

Several academic studies have shown that negative effects of privacy regulations on market concentration can be the result of disproportionate impacts on production costs for smaller firms.⁴⁹

EU law however also aims to ensure a level playing field in which mobile ecosystem developers can develop, deploy or sell their services and products with the EU internal market. Depending on the size and market power of the

actors, EU laws often apply in different ways. Some EU Regulations intend to contribute to the accomplishment of an area where micro, small and medium sized entrepreneurs (SMEs) and large online platforms can operate within the EU internal market under different but fair legal regimes. That is, SME and micro size developers can operate in the internal market with gatekeepers, the latter being subject to more stringent obligations than their smaller competitors (DMA).

Law's enforcement

Laws are enforced by public authorities (regulators, courts, police, supervisory authorities) who are tasked by law to enforce the law against a person's or organisation's will with the aim to regulate conduct of natural and legal persons (including developers) to maintain order, and dispense justice. The most severe measure - law enforcement in terms of fines or closure of an organisation's processing - is often preceded by warnings, penalties and sanctions to ensure compliance by citizens, companies, institutions and others within society. These measures must be laid down by law and can therefore be found in the law applying to the developer. Some laws do not (yet) involve enforcement measures.

⁴⁶ As happened in the context of the Cyber Resilience Directive, see Janssen et al. (2026), available at: <https://zenodo.org/records/22918307>

⁴⁷ Norval, C., Janssen, H., Cobbe, J., & Singh, J. (2021). Data protection and tech startups: The need for attention, support, and scrutiny. *Policy Internet*, 13, 278–299.

⁴⁸ Norval et al. (2021).

⁴⁹ Janssen et al. (2026), available at: <https://zenodo.org/records/22918307>

Risk impact assessments & compliance assessment tools

Importantly, some EU legal frameworks contain or prescribe tools that aim to guide developers through a risk impact assessment procedure. A risk impact assessment measures the potential severity and consequences of a negative event, while a compliance assessment tool checks whether an organisation meets specific applicable laws. A risk impact assessment usually precedes an organisation's check whether laws applying to them are adequately met; in this respect, the two 'tools' are connected (but not all laws contain risk impact assessments).

Risk impact assessment tool

Risk impact assessment procedures aim to raise a developer's awareness of possible risks that their software design might entail for users, for (groups in) society or for others, and aim at a developer's taking appropriate technical and/or organisational measures to mitigate these risks. The GDPR stipulates such a risk impact assessment targeting risks that may come with the processing of a user's personal data which is called a data Protection Impact Assessment, or DPIA (see Article 35 GDPR).⁵⁰

DPIAs is required where processing activities are likely to result in a high risk to individuals' rights and freedoms, and involves the software describing the intended processing, assessing its necessity and proportionality, and identifying measures to mitigate identified risks. The regulatory guidance for the GDPR's DPIA (2017) was developed by the European Data Protection Board (EDPB);⁵¹ yet, the EDPB recently issued a template designed to guide organisations step by step through the DPIA process while allowing developers. This template (and its explanation),

intended to assist organisations in structuring, harmonising and evidencing their DPIA processes, is recommended to all ecosystems developers processing users' personal data.⁵²

If risks occur that could have been mitigated through a DPIA before the product or service was brought to the market, where harms occurred, regulators may issue high fines.⁵³

The AI Act refers to a Fundamental Rights Impact Assessment ('FRIA') which certain developers must perform when they aim to offer high-risk goods or services as defined in the AI-Act.⁵⁴

This FRIA, guidance for which is currently under development, aims to assist developers to guide developers step by step through the FRIA process.

50 Article 29 Data Protection Working Party (this is the predecessor of the current European Data Protection Board), Guidelines on data protection impact assessment and determining whether processing is "likely to result in a high risk" for the purposes of Regulation 2016/679, adopted on 4 April 2017 (DPIA)(WP248Rev.0.1), <https://ec.europa.eu/newsroom/article29/items/611236>

51 Article 4(7) GDPR.

52 Template for Data protection impact assessment and an explanatory document of 10 March 2026; note that these are not the final versions. The EDPB published it for an internet consultation (until 9 June 2026); the finalised version may occur at any moment. See https://www.edpb.europa.eu/public-consultations/template-for-data-protection-impact-assessment_en?page=3

53 Article 83(2)(d) GDPR.

54 Janssen et al. (2026), available at: <https://zenodo.org/records/22918307>

Compliance assessment tool

A typical question usually arising in the context of compliance assessment is whether the applicable laws are being followed in a correct manner. To answer this question, a developer needs to identify, first, what, and whether a law applies to the developer. The law needs to be considered in relation to the purpose the developer pursues, to the goods or services they seeks to offer, the developer terms of service that apply, whether the developer's size or its territory of operation falls within the scope of the law.

In the Table Reflecting how Laws apply to Mobifree developers (Zenodo) we identified in the second row (and in our report) under 'Developer application' whether a specific law would apply to a Mobifree developer (and where possible the specific type of Mobifree developer). We introduced the most outstanding criteria developers should at least consider to examine the law's application to their services or goods. The EU laws involved in our analyses all contain sections where the purpose and scope of the law are described, what the law's key definitions are, who the addressees are, and what compliance aspects and corresponding enforcement rules need to be considered by developers.⁵⁵

For some of the newly adopted laws, the European Commission has already issued high-level background information for organisations about compliance. For some laws regulatory guidance already exists (e.g., GDPR) while for other newly adopted) laws, regulatory guidance may be underway (e.g., AI Act, Cyber Resilience Act).⁵⁶

Once a developer has identified the law's applicability to its operations and services or goods provisions, it should undertake **gap analyses, audit checklists, remediation tracking**, and compliance evidence for regulators such as supervisory authorities (Data Protection Authorities under the GDPR), advisory boards (the Data Act's EU Data Innovation Board), or national authorities (CDSR). At this stage, the developer's main aim is to demonstrate to the relevant regulator that they are responsible for and compliant with the law in question ('accountability').

The law in question must also prescribe that law's purpose, indicating what the law seeks to regulate. Developers also need to identify the law's territorial or material scope. Regarding territory, some laws, only apply whenever a developer offers their services within EU territory. Concerning material scope, other laws only apply to certain types of services such as the Digital Services Act. The Table Reflecting how Laws apply to Mobifree developers aims to give directions on the aspects that developers should consult to determine the law's applicability to their services.⁵⁷

⁵⁵ Janssen et al. (2026), available at: <https://zenodo.org/records/22918307>

⁵⁶ Janssen et al. (2026), available at: <https://zenodo.org/records/22918307>

⁵⁷ Janssen et al. (2026), available at: <https://zenodo.org/records/22918307>

Examples

We recommend developers to first check whether a law applies to them. For instance, to check whether a Mobifree developer's policy and procedures comply with the DMA (Digital Markets Act), a developer should examine whether the DMA applies to their business. From our overview, it appears that the DMA applies to very large sized stakeholders, and not to current Mobifree developers – at least at the moment. The GDPR applies whenever a Mobifree developer processes personal data. Given 'processing' is a very broad notion, developers may process users' personal data.

They may not process personal data if that data is fully anonymised, meaning that a person cannot be re-identified from a data set, even if the data set is combined or enriched with other data. The Equality Access Act applies to Member States in the first place, meaning that Member States ensure enforcement of this Act through national law. If a developer offers goods or services to individuals or groups as defined in the EEA, they should check the law's prescriptions - in both the EEA (as presented in the report) as well as in the national law of the Member State.

Regulatory guidance for compliance assessment as well as mapping the consequences for non-compliance will be helpful for mobile ecosystems developers that are uncertain about the law's application to their operations.

5 Conclusion

This report explained how European mobile stack alternatives can be aligned with the values and needs of different user groups as well as EU laws. By doing so, it contributes to frameworks to build European digital public infrastructure. Our main argument in this report is that the development of more ethical mobile stack layers is not a neutral process and requires development together with different user groups and should be based on EU laws. Mobile stack layers increasingly function as infrastructure – offering essential services for a wide variety of user groups with different needs across nearly all societal sectors. To steer the development of European mobile stacks, broad policy goals and alignment with ‘public values’ risk being too abstract.

Our report proposes a framework to shape all mobile stack layers with two complementary processes: alignment with user values, and compliance with EU law. This ensures that development efforts address mobile software holistically (not focusing on single technologies, but multiple stack layers), that user experiences are taken into account, and that compliance with EU laws can be realised.

Summary of findings

Our test results of existing mobile stack software demonstrate the need for user value alignment.

- Tests with Mobifree software show that different **user groups appreciate many of the tested mobile stack alternatives** and the values they represent. This includes privacy preserving features, transparency features to highlight app tracker software, and software options that function without big tech software.
- **There is a gap between values as abstract concepts and how users perceive them in practice.** Understandability and usability are important factors to invest in to ensure that values built into stack layers (such as privacy- and security-enhancing functionality) is actionable for users and provides a clear added value for them.
- **Users prioritise different values** and can **understand values and associated software functionality in different ways.** Therefore, testing with different user groups is required and user values should be weighed against one another to limit potential harms for some user groups at the benefit of other users.
- **Mobile stack layers are interdependent and can enable or hinder value alignment and legal compliance.** Mobile software is often either built on top of other software, or integrates software (such as middleware). A value that appears to concern one single technology or stack layer, may in fact need to be addressed at another layer of the stack. Value alignment therefore requires a holistic view of the architecture of software and its dependencies to other software.
- Nearly all **user values** identified in the empirical research are already **reflected** in the **EU laws** under review in this report, while they may not get a developer’s attention they deserve. Regulatory guidance should more explicitly and at more detailed level connect user values with laws that ensure users’ rights, and bring obligations to effectuate these rights and the consequences of non-compliance with these rights to the attention of mobile ecosystem developers.

- Developing parties can refer to user values to shape the definition and interpretation of open-natured laws. For example, the way privacy is described in the law may not provide enough guidance for technology design. Developing parties could look at regulatory guidance materials and user values in order to find an appropriate approach to design for privacy.

Our findings have several implications for policymakers and developing parties.

Policymakers should encourage developers and other relevant actors to engage with users and assess how (public) values translate into specific contexts of use. This can help ensure that values are embedded in technologies in ways that are in line with users' actual needs, rather than remaining abstract principles at policy level. Policymakers should also adopt a stack-based view of software infrastructure and governance. If dependencies between stack layers are not accounted for, interventions at one layer may leave underlying dependencies unaddressed. Policy should therefore assess how decisions at one layer affect other layers of the stack and address dependencies across the system. For such a full-stack approach, public rule-setting, regulatory guidance and funding should support the orchestration of development and evaluation processes of multiple stack layers.

Developing parties of alternative mobile stack software should include user testing in a deliberate and structured fashion throughout the entire development cycle. It is important that features are tested for user experience with different groups. This ensures that features promoting certain values are also usable and understandable, and that they do not harm some user groups. The value-sensitive design method followed in this report can be integrated in agile and iterative software development processes of planning, building and evaluation.⁵⁸

⁵⁸ For more explanations of how agile development and value-sensitive design can be matched, see our Guide to Ethical Software Development, available at: <https://publicstack.net/guide-for-co-creation-of-ethical-software>

Literature

Bowker, G. C., and S. L. Star. (1999). *Sorting Things Out: Classification and Its Consequences*. MIT Press: Cambridge. <https://mitpress.mit.edu/books/sorting-things-out>.

Bria, F., Timmers, P., and F. Gernone. (2025). EuroStack - A European Alternative for Digital Sovereignty. *Bertelsmann Stiftung*. <https://www.euro-stack.info/#report>

Dieter, M., A. Helmond, N. Tkacz, F. van der Vlist, and E. Weltevrede. (2021). Pandemic Platform Governance: Mapping the Global Ecosystem of COVID-19 Response Apps. *Internet Policy Review* 10 (3). <https://doi.org/10.14763/2021.3.1568>

Eaves, D., Mazzucato, M., and B Vasconcellos. (2024). Digital Public Infrastructure and Public Value: What is 'public' about DPI? *Institute for Innovation and Public Purpose*. Available at: <https://www.ucl.ac.uk/bartlett/publications/2024/mar/digital-public-infrastructure-and-public-value-what-public-about-dpi>

Edwards, P. N., Jackson, S. J., Bowker, G. C., and C Knobel. (2007). Understanding infrastructure: dynamics, tensions, and design. Working paper. Available at: <http://deepblue.lib.umich.edu/handle/2027.42/49353>

European Union. (2025). The Draghi Report on EU Competitiveness. *Publications Office of the European Union*. https://commission.europa.eu/topics/competitiveness/draghi-report_en

Friedman, B, and D. G. Hendry. (2019). Value Sensitive Design: Shaping Technology with Moral Imagination. The MIT Press. <https://doi.org/10.7551/mitpress/7585.001.0001>

Gerlitz, C., Helmond, A., Nieborg, D. B., and F. N. van der Vlist. (2019). Apps and Infrastructures – a Research Agenda. *Computational Culture*, 7. <http://computationalculture.net/apps-and-infrastructures-a-research-agenda/>.

Heinich, N. (2020). A Pragmatic Redefinition of Value(s): Toward a General Model of Valuation. *Theory, Culture & Society* 37 (5): 75–94. <https://doi.org/10.1177/0263276420915993>

Janssen, H, Valk, E, Toepoel, I and Cordoba Lenis, C. (2026). Towards a Compliant Mobile Ecosystems Design: Exploring laws applying to apps, app stores, cloud and operating systems

Keller, P., and J. Krewer (Open Future). (2024). Mapping the Debates About Strengthening Europe's Digital Infrastructure.

Larkin, B. (2013). The Politics and Poetics of Infrastructure. *Annual Review of Anthropology* 42 (1): 327–43. <https://doi.org/10.1146/annurev-anthro-092412-155522>

Le Dantec, C. A., Shehan Poole, E., and Susan P. Wyche. (2009). Values as Lived Experience: Evolving Value Sensitive Design in Support of Value Discovery. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1141–50. <https://doi.org/10.1145/1518701.1518875>

Norval, C., Janssen, H., Cobbe, J., & J. Singh. (2021). Data protection and tech startups: The need for attention, support, and scrutiny. *Policy Internet*, 13, 278–299.

Ovadya, A., Redman, K., Thorburn, L., et al. (2025). Position: Democratic AI Is Possible. The Democracy Levels Framework Shows How It Might Work. *Proceedings of the 42nd International Conference on Machine Learning*, 81930–61. <https://proceedings.mlr.press/v267/ovadya25a.html>

Rajendra-Nicolucci, C., and M. Samdub. 2024. What Is Digital Public Infrastructure? Towards More Specificity. *Tech Policy Press*. <https://techpolicy.press/what-is-digital-public-infrastructure-towards-more-specificity>

Star, S. L., and K. Ruhleder. 1996. Steps Toward an Ecology of Infrastructure: Design and Access for Large Information Spaces. *Information Systems Research* 7 (1): 111–34. <https://doi.org/10.1287/isre.7.1.111>

Colofon

Publication

Waag Futurelab 5 October 2026

Authors and editors

Danny Lämmerhirt, Heleen Janssen, Isadora Dullaert, Etienne Valk, Iris Toepoel, Christian Cordoba Lenis, Natalie Sahusilawani, Bente Zwankhuizen, Sophie Luykx, Bas van Druten

Contact details

Waag Futurelab

Sint Antoniesbreestraat 69
1011 HB Amsterdam
The Netherlands