MOBILE APPS AND GDPR ISSUES

Alexandru TĂBUŞCĂ [1*]
Silvia-Maria TĂBUŞCĂ [2]
Gabriel Eugen GARAIS [3]
Alexandru ENĂCEANU [4]

ABSTRACT

As the Internet has really became a must for all modern societies, there are a lot of modifications that appear, somewhat regularly, related to the use of services over this networking infrastructure. During the last years we have seen a surge in the use of mobile apps, especially after the invasion of smartphones (which actually surpassed the "standard", feature phones in most markets). The ubiquitous smartphone is a permanent companion for most people, especially young one but the usage levels is at very high levels in middle-aged category too, and the internet connection is now a given-fact for a smartphone user. Almost all software vendors today provide a mobile app to complement, or in a few cases even replace, the traditional software they create. All those mobile apps have the main advantage of being right at hand, on our smartphones, being able to fulfill a need (at least at a certain level) without the actual use of a traditional PC, either desktop or laptop. Most, if not all, of these mobile apps require us to grant them access to different sets of personal information. These data have now to be filtered through the requirements of the latest EU data regulation legislation, which is enforced from May 2018. The GDPR documents provide very specific and detailed requirements that all personal data processor must obey from that moment on.

KEYWORDS: *Mobile apps, gdpr, data protection, rdf, xcode, swift*

1. INTRODUCTION

The Internet, as the network of all networks, is being considered now, within the advanced and even emerging societies, as a normal standard, a “right” in its own. The hastiest society in legally changing the status of the internet, of the access to internet to be more specific, was Finland. Finland became, in 2010, the first country to insert into the legislation the lawful right of their citizens to have access to an internet connection [1],

[1*] corresponding author, Associate Professor PhD, Faculty of Computer Science for Business Management, Romanian-American University, Bucharest, tabusca.alexandru@profesor.rau.ro

[2] Lecturer PhD., Romania-American University, laura.maniu@gmail.com

[3] Lecturer PhD, Faculty of Computer Science for Business Management, Romanian-American University, Bucharest, garais.gabriel@profesor.rau.ro

[4] Lecturer PhD, Faculty of Computer Science for Business Management, Romanian-American University, Bucharest, enaceanu.alexandru@profesor.rau.ro

namely a 1Mbps connection at least (from July 2010) and at least 100Mbps connection from 2015. The internet providing companies (ISPs) have become some of the most important and relevant companies on the modern markets of today. The ISPs, grown from the street/block-level to huge giants, have also become very much involved, directly and indirectly, into every layer of the society fabric: providing services, ensuring connectivity and communications, and even powerful CSR campaigns [2].

This ubiquitous internet “monster” is now a must for all modern societies and especially for the young(er) generations. This fact adds quite a lot of modifications that appear, related to the use of services over this networking infrastructure – being able to get to your customers very fast and actually without a geographical barrier gives online players the opportunity to change approaches, techniques and strategies way faster than before. Over the last years we have all witnessed a cascade of mobile applications, especially after the invasion of the smartphones (which actually surpassed the "standard", feature phones, in most markets). The omni-present smartphone is a permanent companion for most people - especially young ones, but the usage level is at very high percentage in middle-aged category too - and the internet connection is now a given-fact, for granted, to a smartphone user. Almost all software producers today provide a mobile application to complement, or in a few cases even replace, the traditional software they create. All those mobile apps have the main advantage of being right at hand, on our smartphones, being able to fulfill a need (at least at a certain level) without the actual need and use of a traditional PC, either desktop or laptop.

Most, if not all, of these mobile apps require us to grant them access to different sets of personal information. The data has to be somehow transferred to a server location – and we will present a study case based on using RDF data transfer packaging – and be processed – and we will present the case of a real application developed for the Apple environment.

All these (possibly huge) amounts of data have now to be also “filtered” through the requirements of the latest EU data regulation legislation, which is enforced from May 2018. The EU’s GDPR documents provide very specific and detailed requirements that all personal data processors must obey from that moment on.

The EU’s GDPR holds answers for the most relevant questions related to collecting, storing and processing of data related to personal information: what, when, how to do all those steps.

2. DATA TRANSFERS THROUGH RDF STANDARD

Using RDF technology in context of GDPR can easy the relationships between data transfer or storing and their meaning.

RDF stands for Resource Description Framework and is a framework for defining information and their relation about resources. Resources can be documents, persons, objects, and abstract definitions. RDF is meant for implementations in which data on the web must be processed by software programs, not only be displayed to users. RDF offers a standardized framework for presenting this information, so it can be transferred between websites or software with no misleading meaning. Being a standard framework, developers can use available definitions of standard RDF. Exchanging information through this framework offers the possibility to put data available for other applications than the original ones.

The RDF Data Model consists of simple statements with the following structure:

<subject> <predicate> <object>.

The RDF statement defines a relationship in relation with two resources. A subject and an object define the structure of the resources being in relation and a predicate defines the character of connection. The relationship is defined in a straight path from subject to object and is characterized as an RDF property. RDF statements structure is called triples because of the three properties they use as in the example below.

<Gabi> <is a> <person>.
<Gabi> <is a friend of> <Ana>.
<Gabi> <is born on> <the 1st of December 1918>.
<Gabi> <is interested in> <the ArtOfWar>.
<the ArtOfWar> <was created by> <SunTzu>.
<the documentary 'Thinking about the Art of War'> <is about> <the ArtOfWar>

The triples can be structured as a connected graph which consist of nodes and arcs. The nodes of the triples in the graph are structured by subjects and objects and the predicates form the arcs. The Figure 1 presents the resulted graph using the previous example.

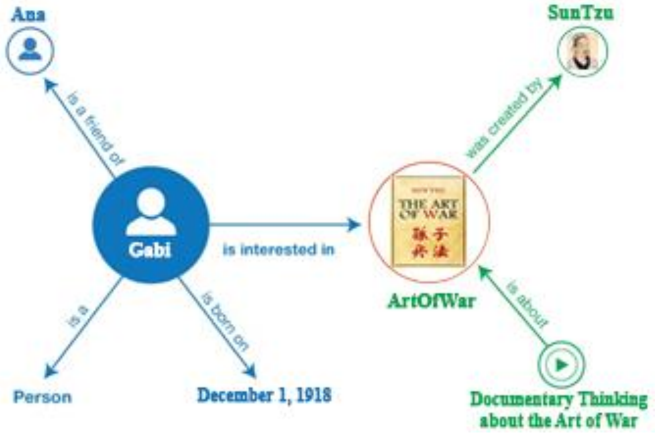


Figure 1. Graph of the sample using subject, predicate and object [1]

RDF offers a structure to define RDF statements in multiple graphs and relate graphs with an International Resource Identifier (IRI). Multiple graphs are a recent extension of the RDF data model. The RDF data model was extended with a structure of multiple graphs that is closely aligned with SPARQL query language which is a recursive acronym for SPARQL Protocol and RDF Query Language.

[1] https://dvcs.w3.org/hg/rdf/raw-file/default/rdf-primer/index.html

Multiple graphs in an RDF document, structure an RDF dataset which can have multiple named graphs and at maximum one unnamed graph. For example, the statements in first RDF example could be grouped in two named graphs. A first graph could be presented by a social networking website and identified by http://rau.ro/gabi:

> <Gabi> <is a> <person>.
> <Gabi> <is a friend of> <Ana>.
> <Gabi> <is born on> <the 1st of December 1918>.
> <Gabi> <is interested in> <the ArtOfWar>.

The second graph provided in this example is from Wikidata and has the following url:

http://www.wikidata.org/entity/xxx5

> <SunTzu> <is the creator of> <the ArtOfWar>.
> <the documentary 'Thinking about the Art of War'> <is about> <the ArtOfWar>

The next example is stated for an unnamed graph and is structured on two triples with the graph name <http://rau.ro/gabi> as subject.

The triples are related to the publisher and license information with this IRI graph:

> <http://rau.ro/gabi> <is published by> <http://rau.ro>.
> <http://rau.ro/gabi> <has license> <http://rau.ro/publiclicense/1.0/>.

Figure 2 illustrates the sample dataset which adds more structured meaning to the example.

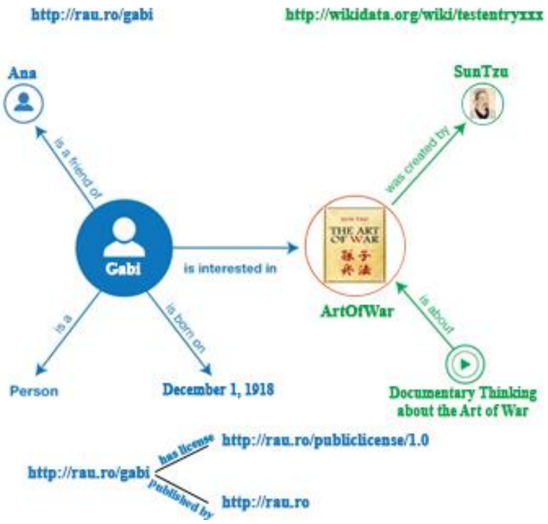


Figure 2. Multiple Graph statement of the sample using subject, predicate and object [1]

[1] https://dvcs.w3.org/hg/rdf/raw-file/default/rdf-primer/index.html

In application implementations RDF is usually inserted in relation with vocabularies or other semantic notation types that provide information about the resources. As for semantic analysis using vocabularies, N-Triples structures main syntax for writing RDF triples. N-Triples is type of Turtle syntax which extends this basic syntax to improve readability.

The N-Triples provides a simple plain-text way for serializing RDF graphs. The graph in Figure 1 can be illustrated in N-Triples as in the following example:

```
01 <http://rau.ro/gabi#me> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://xmlns.com/foaf/0.1/Person>.
02 <http://rau.ro/gabi#me> <http://xmlns.com/foaf/0.1/knows> < http://rau.ro/ana#me>.
03 <http://rau.ro/gabi#me> <http://schema.org/birthDate> "1918-12-
01"^^<http://www.w3.org/2001/XMLSchema#date>.
04 <http://rau.ro/gabi#me> <http://xmlns.com/foaf/0.1/topic_interest>
<http://www.wikidata.org/entity/xxx1>.
05 <http://www.wikidata.org/entity/xxx2> <http://myexampleurl.org/title> "ArtOfWar".
06 <http://www.wikidata.org/entity/xxx3> <http://myexampleurl.org/creator>
<http://myexample2.org/SunTzu>.
07   <http://myexample3.org/xxx4><http://myexampleurl.org/subject>
<http://www.wikidata.org/entity/xxx5>.
```

The example presents many lines which represents individual triples. IRI-s are enclosed in standard tag brackets (< >). The end of a triple signalized at each end of a the line with a ".". In line 3 we see an example of a literal which are basic values that are not IRIs, in this case a date. Each literal has an attached datatype using the ^^ delimiter. The date description uses standard datatype date of XML Schema.

In Figure 3 the resulting triples are presented using the Schema vocabularies of RDF that define in a meaningful structure the graph of statements which define and structure the semantic relation between subject, predicate and object.

The RDF standard is still improving and developing constantly. Its use does not only stop at describing data for transfer processes. The RDF might also be used for describing networks [3] and other complex build-ups.

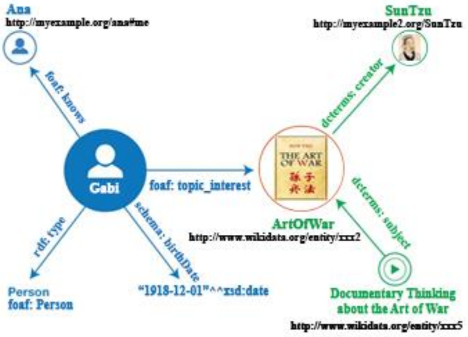


Figure 3. Illustrating the N-Triples that serialize RDF graphs[1]

[1] https://dvcs.w3.org/hg/rdf/raw-file/default/rdf-primer/index.html

The explained Graphs structured through examples of RDF vocabularies and semantics represent a technical representation of how data is defined and provide a way of exchanging information between applications. The using of RDF way of exchange and defining of information in the context of GDPR is important because it provides a type of structuring information based on its meaning and relationships in such a way to avoid any violation of data protection that reaches public websites and gathering of personal sensitive information.

3. DATA TRANSFER APPLICATIONS IN XCODE (SWIFT) PROJECTS

After having different types and amounts of data ready to be transferred and processed, we actually need an application to do it. The iOS development has several principles which represent, in the same time, both a “pro” and a “con” sides. The iOS programming principles [4] help with providing sturdy applications but they also restrict some other capabilities which could be found in other programming environments.

One of the most challenging, but also rewarding, environments for mobile applications is the Apple one. We shall briefly present an implementation of a procedure for taking in and processing remote data – a real mobile application implementation which uses different technologies for providing a scheduling/presentation app for keeping track of a certain public event.

Getting and parsing structured data from remote servers can be very tough in Xcode (Swift), but due to the Alamofire[1] and Swiftyjson[2] libraries, it becomes an easy task.

In order to install Alamofire + Swiftyjson you need to install cocoapods[3] first:

```
gem install cocoapods
```

and then pod setup.

To add Alamofire and SwiftyJSON Pods to the project go to the directory containing the AlamofireSwiftyJSONSample:

```
cd ~/Path/To/Folder/Containing/AlamofireSwiftyJSONSample
```

and give a pod *init* command.

Then, open the podfile:

```
open -a Xcode Podfile
```

Edit the podfile and add the following targets:

```
target ‘AlamofireSwiftyJSONSample’ do
   pod ‘Alamofire’
   pod ‘SwiftyJSON’
end
target ‘AlamofireSwiftyJSONSampleTests’ do
```

[1] https://github.com/Alamofire/Alamofire

[2] https://github.com/SwiftyJSON/SwiftyJSON

[3] https://cocoapods.org/

```
end
target ‘AlamofireSwiftyJSONSampleUITests’ do
end
```

Go to the terminal mode and give a pod install.

After the installation has been successfully done, import it in Xcode by using the:

```
import UIKit
import Alamofire
import SwiftyJSON
class ViewController: UIViewController {
}
```

Next, use the *Alamofire.request* method to get the response from the server:

```
Alamofire.request("https://hostname/mysjsonfile").responseJSON {(responseData)
-> Void in
     if((responseData.result.value)!= nil) {
        let swiftyJsonVar = JSON(responseData.result.value!)
        print(swiftyJsonVar)
     }
 }
```

Next step is to populate the variable and outlets:

```
@IBOutlet var tblJSON: UITableView!
var arrRes = [[String:AnyObject]]() //Array
```

The Alamofire 4.0 script will look as following:

```
override func viewDidLoad() {
    super.viewDidLoad()
    Alamofire.request("https://hostname/mysjsonfile").responseJSON { (responseData) ->
    Void in
       if((responseData.result.value) != nil) {
          let swiftyJsonVar = JSON(responseData.result.value!)

          if let resData = swiftyJsonVar[“myvariable”].arrayObject {
             self.arrRes = resData as! [[String:AnyObject]]
          }
          if self.arrRes.count > 0 {
             self.tblJSON.reloadData()
          }
       }
    }
}
```

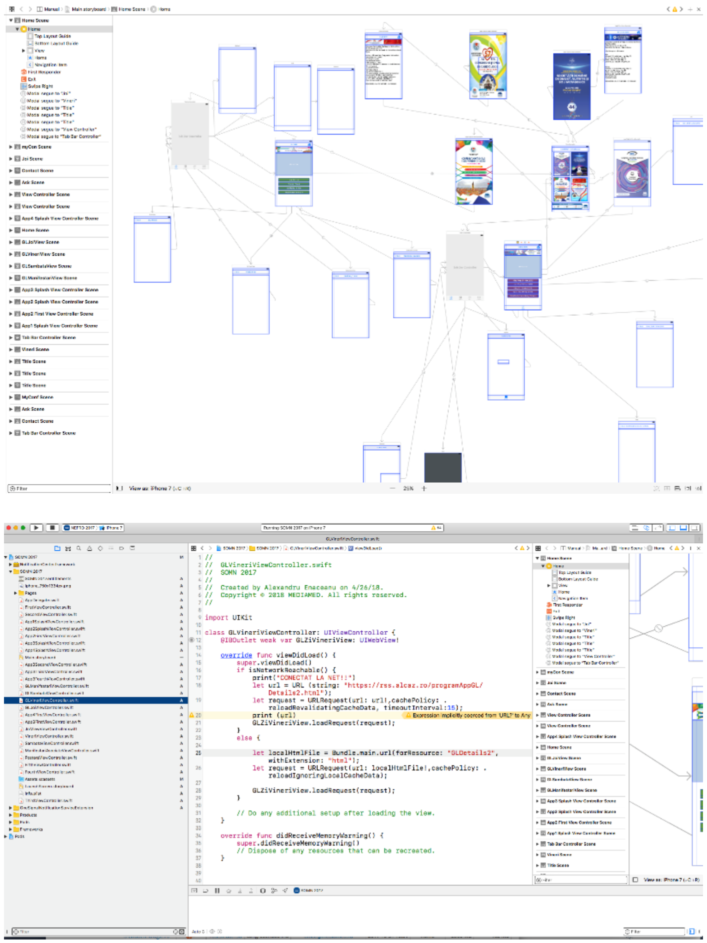
Final step is to delegate the methods of the table:

```
func tableView(tableView: UITableView, cellForRowAtIndexPath indexPath:
NSIndexPath) -> UITableViewCell
{
let cell: UITableViewCell = tableView.dequeueReusableCellWithIdentifier("jsonCell"
)!
var
 dict = arrRes[indexPath.row]
      cell.textLabel?.text = dict[
“myvar1”] as? String
      cell.detailTextLabel?.text = dict[
“myvar2”] as? String
return
 cell
  }

func tableView(tableView: UITableView, numberOfRowsInSection section: Int) -> Int
 {
return arrRes.count
 }
```

We have used the above procedures to implement the remote synchronization of the example mobile application, entitled EvMedicale and available in the AppStore, to deliver the daily conference schedule (hours, rooms, conference title, moderators etc.).

The application structure and one ViewController example can be viewed in the below xCode screenshots:



```
//
//  GLVineriViewController.swift
//  SOMN 2017
//
//  Created by Alexandru Enaceanu on 4/26/18.
//  Copyright © 2018 MEDIAMED. All rights reserved.
//

import UIKit

class GLVineriViewController: UIViewController {
    @IBOutlet weak var GLZiVineriView: UIWebView!

    override func viewDidLoad() {
        super.viewDidLoad()
        if isNetworkReachable() {
            print("CONECTAT LA NET!!")
            let url = URL (string: "https://rss.alcaz.ro/programAppGL/
                Detalii2.html");
            let request = URLRequest(url: url!,cachePolicy: .
                reloadRevalidatingCacheData, timeoutInterval:15);
            print (url)
            GLZiVineriView.loadRequest(request);
        }
        else {

            let localHtmlFile = Bundle.main.url(forResource: "GLDetails2",
                withExtension: "html");
            let request = URLRequest(url: localHtmlFile!,cachePolicy: .
                reloadIgnoringLocalCacheData);

            GLZiVineriView.loadRequest(request);
        }

        // Do any additional setup after loading the view.
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }
```

⚠ Expression implicitly coerced from 'URL?' to Any

Figure 4. Mobile application structure and ViewController example

In the below figure 5 we can see the actual, real-life screenshots of the application, from the Apple AppStore:

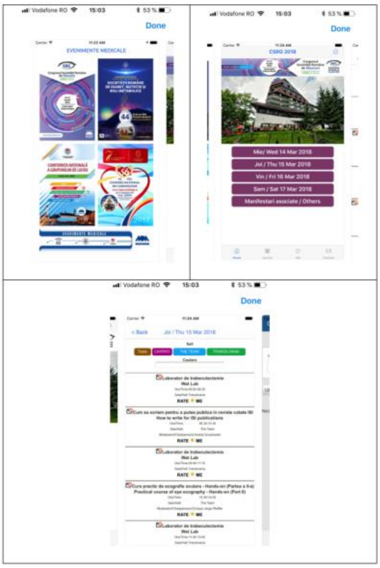


Figure 5. Mobile application available inside the AppStore

4. GDPR IN THE CONTEXT OF DATA TRANSFERS AND MOBILE APPS

When transferring data (possibly with RDF standards) to mobile applications (like the case study presented above) we must take into consideration the issue of the personal information. Even more than before, the protection of personal information related data is now more in focus. The European Union has settled on a new piece of legislation, the GDPR[1] - General Data Protection Regulation. This is the most important update of the data privacy regulations within EU for the last 20 years.

The GDPR actually replaces a previous EU directive, Data Protection Directive 95/46/EC from 1995. EU has studied, at different levels, three different version for this initiative:

- ✓ 2012 - January 25th, initial proposal for updated data protection regulation by the European Commission
- ✓ 2014 - March 12th, the European Parliament approved its own version of the regulation in its first reading
- ✓ 2015 - June 15th, the Council of the European Union approved its version in its first reading, known as the general approach, allowing the regulation to pass into the final stage of legislation known as the “Trilogue”

After all procedures have been made, the EU Parliament and the EU Council agreed on the final text of the act, which was signed in January 2016. In April of 2016 bot the Parliament and the Council have officially adopted the act.

The regulation will be enforced in May 2018, 20 days after being published in the EU Official Journal.

[1] https://www.eugdpr.org/

One way of minimizing the security risks related to Data Protection, for personal information especially, is the concept known as “privacy by design”. The European Union General Data Protection Regulation (EUGDPR) states that “*...the principle of data protection by design require data protection to be embedded within the entire life cycle of the technology, from the very early design stage, right through to its ultimate deployment, use and final disposal. This should also include the responsibility for the products and services used by the controller or processor.*”

This regulation procedure thus states that quite a strict process needs to be implemented, from the early stages of development of the applications and up to the final stage of its implementation and use. This new “privacy by design” concept was in fact required based on the consideration that it is able to provide a better environment through the following:

- Reducing risks related to mobile applications being considered as not compliant with the latest privacy acts (namely GDPR)
- Reducing potential legal issues derived from activities such as cyber-crimes, because security problems should be reported to the legal authorities/entities only if the controller of the personal information data is not capable of demonstrating the fact that he took all necessary precautions for assuring a high level of security for the data processing procedures.
- Excluding other potential legal problems related to the electronically collected and processed data by anonymizing as much as possible all non-critical information (from the services provided point of view).

This concept of “privacy by design” is not solely used and requested by the EU. Both US enforcing institutions, such as the US Federal Trade Commission (FTC), and UK’s (OFCOM) has imposed similar requirements even before GDPR.

The EU’s GDPR brings a series of new elements that are important for the legal aspect over the matter at hand:

- Possible fines for privacy related problems are increased to up to 4% of the global turnover of the guilty entity
- A much probable possibility for the customers to claim compensations (quite similar to several US legal possibilities) because the GDPR act moves the proving requirement part from the customer to the investigated party. Namely, in some cases, it is not the customer who should demonstrated something that was wrong, but the company has to demonstrate that it did not do something wrong.
- A much higher possibility for shareholders to lay claims from the executive branch, based on the possible huge fines related to breaking of GDPR regulations.

After the new regulations, companies that want to be active within the EU “e-space” must implement security measures in order to reduce the cyber-attacks risks and must prove compliancy with the concept of ordinary diligence. The most important steps that a company should take in order to answer these issues are:

- Adopting a clear cyber-security policy
- Investing into a cyber-risk insurance
- Using privacy by design paradigm
- Employing a DPO (Data Protection Officer)

The newly created, by GDPR, Data Protection Officer position is not meant to be only a guideline proposal, but in some cases, it becomes mandatory. For example, if the most important activity field of a certain company is based on data processing operations which require permanent monitoring of data subjects on large scale – e.g. companies like banks, telecom integrators, fintech/insurance companies, medical practices – or, if the most important activity of that certain company is based on processing large-scale amounts of special data – e.g. biometric, medical data etc. – then, the company is actually forced to hire or appoint a DPO. This DPO position must be able to perform its tasks independently, without following internal instructions on how to do its job (at data security level, of course it has to abide by a clothing protocol for example – if the company has one). The Data Protection Officer, according to the EU's GDPR, should act as to:

- Collect information, in order to identify the processing activities
- Analyze and check the law compliance of the processing activities
- Inform, advise and afterwards recommend to the company's executives on security/privacy problems linked to the processing activities
- Cooperate with the legal authorities, as being the company's contact point on problems linked to data processing.

Nevertheless, we have to mention the Data Protection Officer, at the end of the day, still has mostly a supportive role. The executive branch of the company, the decision maker, is the one that finally takes the decision to follow the Data Protection Officer's recommendations or not. On the other hand, if something wrong happens and the DPO did not make a previous recommendation (based on the GDPR regulations being potentially breached), then he is the one that bears the responsibility.

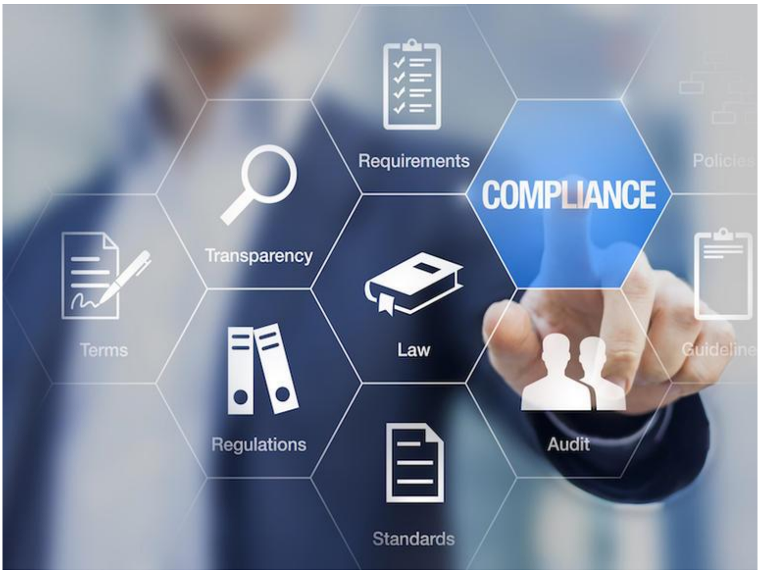


Figure 6. Main EU's GDPR areas

As final summary, we can use figure 6, above, to point out the main areas covered by the newly enforced GDPR act: terms of contracts, transparency related to data requirements, strictly following the enforced regulations, check if all regulation requirements are implemented, check all law related approaches, meet all standards of the law, use a regular schedule for audit of the security related policies.

REFERENCES

[1] Tabusca, Silvia Maria - The Internet Access as a Fundamental Right; published in Journal of Information Systems and Operations Management, Vol.4. No.2 / 2010, pp 206-212, ISSN 1843-4711.

[2] Edu, Tudor; Negricea, Iliuta Costel - CSR Market Positioning Constructs: From Planning to Action. Evidence from Romanian Internet Service Providers; published in book The Dynamics of Corporate Social Responsibility, 2017, pp 117-137, ISBN 978-3-319-39089-5.

[3] van der Ham, Jeroen J.; Dijkstra, Freek; Travostino, Frank; Andree, Hubertus; de Laat, Cees T.A.M. - Using RDF to describe networks; published in Future Generation Computer Systems, 2006, pp.862-867, ISSN: 0167-739X.

[4] Woodhouse, Clare - 8 Principles of iOS Development to Know Before Starting Your App, https://www.upwork.com/hiring/mobile/hiring-an-ios-app-developer/, published on: September 15, 2015; last access: May 01, 2018