European Commission

JRC TECHNICAL REPORT

Update of Risk Data Hub software and data architecture

Software solutions for Disaster Risk Management

2020

Antofie, T., Luoni, S., Eklund, L, G., Montserrat, M, F.,

Joint Research Centre

EUR 30065 EN

Contact information
Name: Antofie Tiberiu-Eugen
Address: Via E. Fermi, 2749, TP 267, I – 21027, Ispra (VA), Italy
Email: Tiberiu-Eugen.ANTOFIE@ec.europa.eu
Tel.: +39 0332 78 9363

EU Science Hub
https://ec.europa.eu/jrc

JRC119500

EUR 30065 EN

PDF	ISBN 978-92-76-15386-3	ISSN 1831-9424	doi:10.2760/798003

Luxembourg: Publications Office of the European Union, 2020

How to cite this report: Antofie, T, E., Luoni,, Eklund, G., Marin Ferrer, M., *Update of Risk Data Hub software and data architecture,* EUR 30065, Publications Office of the European Union, Luxembourg, 2020, ISBN 978-92-76-15386-3, doi:10.2760/798003, JRC119500.

Joint
Research
Centre

Contents

Authors

Tiberiu Antofie, European Commission, Joint Research Centre (JRC), Ispra, Italy

Stefano Luoni, Unisystems Sarl, External consultant of the European Commission

Montserrat Marin Ferrer, European Commission, Joint Research Centre (JRC), Ispra, Italy

Gustav Eklund, European Commission, Joint Research Centre (JRC), Ispra, Italy

Abstract

The Risk Data Hub is an initiative of the Disaster Risk Management Knowledge Centre (DRMKC) and consists of a publicly available web-GIS platform intended to improve the access and sharing of curated European-wide risk data and methodologies for fostering Disaster Risk Management (DRM) related actions.

The implementation of the concept is made of multiple steps, including the definition of type of analysis to be presented, the design of methodologies to compute data needed, the design of the database architecture and software tools and, finally, the development of the software.

This document, as un update of the first technical report (JRC114712) on software and data architecture, will focus on the design, starting from a high level analysis of the business needs, going to the explanation of the solutions proposed, considering previous works in the topic of Disaster Risk Management and showing how the existent Loss Database architecture has been reviewed to fit the requirements of a complex and multi-context application.

For a more generic overview of the concept of the DRMKC Risk Data Hub, the reader can consult the report "*The Disaster Risk Management Knowledge Centre – Risk Data Hub: Vision Paper & roadmap"*, European Commission, Ispra, 2019, JRC119384.

.

1 Introduction

Despite the great number of projects developed in the context of Disaster Risk Management, there are no widely shared resources and common methodologies used for the analysis of disaster risk data. Moreover every country has different level of maturity and expertise, having in some cases its own single-hazard (rarely multi-hazard) databases and institutional capacities, with different levels of usage and effectiveness.

With the DRMKC Risk Data Hub the European Commission wants to offer a common platform to access data and methodologies, granting a more harmonised and facilitated approach to the Risk Management for the end-user.

The main objective of this report is to have an overview on the updates of the technological developments of the platform both, at the level of visualisation but also at the level of data structure and analysis.

To understand what this application is about, we can start with a couple of considerations.

Addressing risk data on a GIS Web-platform means, in the first place to consider the relation hazard-exposure-vulnerability. The Risk Data Hub has a multi-hazard approach, implementing methodologies that helps displaying and also defining the hazard as single layer and as multiple comparable layers (e.g. implying an alignment of methodological approaches and data used for disaster risk assessment across different hazards). While up to present, the datasets introduced are related to natural hazards, the technological and man-made hazards are currently being planned, with the final aim of having a complete mapping of risk, also including assessments for both direct and indirect impacts.

The DRMKC Risk Data Hub is also a multi-context platform, as it can be used to access exposure and vulnerability assessments (useful for risk reduction), as well as historical events catalogue (post-event) to analyse empirical vulnerabilities and trends. This means that on a single platform the user has the possibility to discover exposed and vulnerable areas for every hazard, verify and compare risk against past impacts, perform statistical analysis, find trends following Sendai indicators, check potential eligibility for solidarity fund requests and more.

Risk Data Hub wants to be a "second house" for research results, satisfying the need to make them accessible. It is a collaborative platform where, starting from a strong partnership across different scientific groups dealing with different hazards, the scientific information is harmonised and translated into evidences for the end-user. This purpose may be clarified by defining input and output for this platform.

Input is granted by scientific partnerships trough a collaborative network created for discovering already existing data, actions, and practices for DRM.

Output consists of different analysis performed on available data. The Risk Data Hub is developed as a decision support system that integrates spatial data along with statistical analysis. This helps decision makers have an indication for time and spatial coverage of the economic damages and human losses across Europe from hazardous events, upon which consistent decisions can be taken

A more complete explanation of methodologies developed for Risk Data Hub and its relation to policies is included in "Risk Data Hub - web platform to facilitate management of disaster risks" JRC technical report.

2 Main challenges and solutions identified

The DRMKC Risk Data Hub is the convergent point for a number of activities lead by EC over the last years aiming from one side at establishing a common understanding of the value and the methodological approach regarding damage and loss data[1], while for the other side describing the different methodologies for risk assessment adopted by the different hazards and threats[2,3]. The Risk Data Hub is the practical implementation of all this information.

Before starting any implementation, all concepts, methodologies and business needs have to be translated into technical requirements. This chapter contains a descriptive and simple explanation of such analysis process.

2.1 Better characterization of hazardous events, using phenomena

Assessing damages and losses requires defining an event, which simply tells us where and when damages are located and what caused them (e.g. a flood, or an earthquake).

Understanding what an event is in the real world is quite simple: the earthquake happened in August 2016 in central Italy is an event, while floods happened in August 2002 in central Europe are another event – actually, a set of different events in Risk Data Hub, as the platform identifies a distinct event for each country, but the general idea of an event is still clear.

An **Event** is defined into Risk Data Hub considering **Hazard, Begin Date, End Date** and **Country** as unique together. Using the country level for localising an impact is clearly not enough, as we typically expect to get some more detailed information and that's where phenomena come in place: any impact bound to a specific location is stored, within an event, by a **Phenomenon**. This approach is coherent with the hierarchical representation of the phenomenon used in Global Disaster Alert and Coordination System (GDACS). For GDACS the concept of episodes is utilized and it is the equivalent of the RDH events. Likewise, episodes are individual measures of a phenomenon at a given time by a given organisation.

The initial implementation of phenomena in the RDH database included columns for **Event ID, Begin Date, End Date** and **Location**, where the latter may be any administrative unit within the country, from NUTS2 level to LAU (Local Administrative Unit) level. This way it was possible to store different impacts and group them into a bigger event which may refer to, for example, an atmospheric event as a whole.

As of early 2019, the visualization of phenomena on RDH maps was limited because of lack of data: basically, only a list of locations was presented for each event, as the example datasets collected did not contain any details of impact for phenomena within events, but only a value for the bigger event.

The opportunity to represent more details emerged after starting collecting geographic extent of areas burned by Forest Fires: following the above logic for defining events, it appeared possible to translate each piece of area affected by an hazardous event to a phenomenon, having not only the numeric value of the extension of the area, but the complete spatial information, including the geometry.

This is where another problem showed up: as the administration units associated to data were NUTS3 and most of the single burned area were relatively small, the model used for phenomena was not fitting data anymore, as many cases of separate burned areas occurred in the same administration unit and same date were identified (see chapter 4 for more details on changes applied to data models).

So, in the end, the visualization of phenomena on the web platform required a preliminary action to be done, which consisted of changing the data structure in a way that allowed to manage this particular complexity.

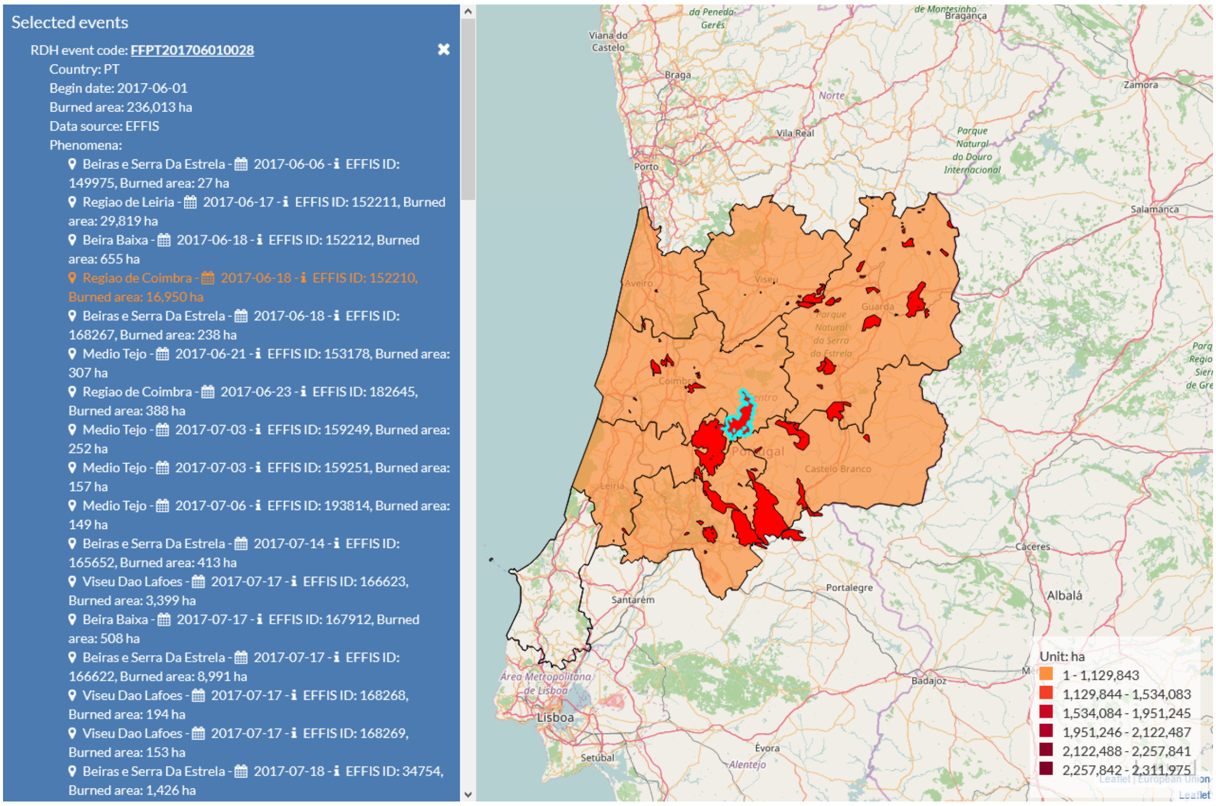
Figure 1 shows all burned areas observed during summer 2017 in a particular region of Portugal (Centro). This is a demonstration of the concept of multiple phenomena bound to an event, which in this example is an aggregation of fires happened in the summer season of 2017. Each phenomenon is selectable, to clearly spot its extent on the map.

[1] https://drmkc.jrc.ec.europa.eu/partnership/Science-Policy-Interface/Disaster-Loss-and-Damage-Working-Group

[2] https://drmkc.jrc.ec.europa.eu/portals/0/Knowledge/ScienceforDRM/ch02/ch02_intro.pdf

[3] https://drmkc.jrc.ec.europa.eu/knowledge/SCIENCE-FOR-DRM/Recommendations-for-National-Risk-Assessment-for-Disaster-Risk-Management-in-EU

Figure 1. Event and related phenomena assessing damages and losses



Source: Risk Data Hub, 2019

The approach of using such Event/Phenomenon data model was introduced at first by the FloodCat database, which was designed to fulfil the Floods Directive Reporting and adopted by Italian Civil Protection. The 2016 study "*Development of EU harmonized services for recording flood events and associated damages data accommodating Floods Directive and towards the implementation of Sendai Framework*", delivered by the Cima Research Foundation under the umbrella of DRMKC support system[4], started then to propose an EU wide usage of FloodCat considering both the implementation of the Sendai Framework and the INSPIRE Natural Risk Zones technical guidelines.

Risk Data Hub shall be considered as the natural prosecution of these activities, with the added ambition of getting to a unique model and to standard methodologies for recording and analysing EU wide, multi-hazard and multi-sector Loss Data and Risk Data.

2.2 Displaying data from multiple sources

As explained in previous reports, Risk Data Hub collects data from **multiple sources**, with the purpose of offering a complete and comparable view of existing disaster data.

While attribution of a source was always done for data introduced into the system, this information was simply reported as part of metadata, so not used to perform any logic.

So, what happens when we manage to collect data from multiple sources for one, or a set of the same past event(s)? The logic implemented by RDH to identify an event and avoid duplications is not part of this document, as it was previously reported. Speaking about data visualization, it is true that the first version of data models didn't allow to manage very well this need of displaying different values for an event.

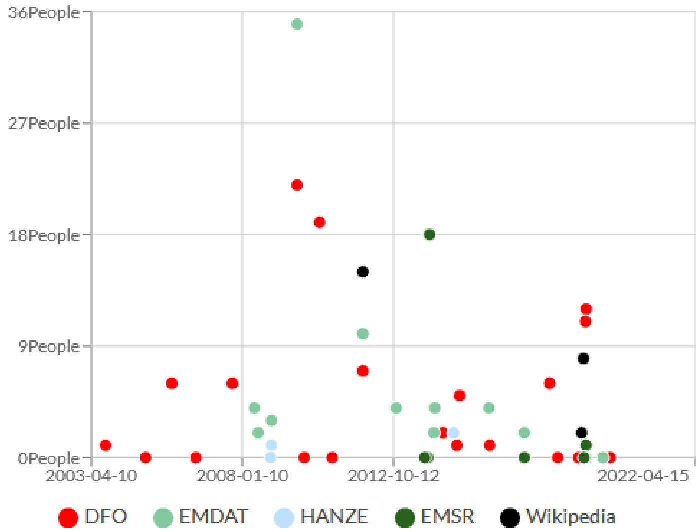
In case of multiple values are stored for a single event, one can think of either listing each event with all related records, or picking a single value at once. This two approaches are both valid, but depending on the type of visualization, one option would be better than the other.

[4] https://drmkc.jrc.ec.europa.eu/innovation/SupportSystem

The scatter chart illustrated in figure 2 shows the distribution over time of a set of events (fatalities from River Floods in this example), where each data source is reported in a different colour. For this particular visualization, it's clear that displaying multiple values for a single event would make such chart a real mess.

In the end, the approach used was to put data sources in a **priority list** and, for each event, to pick the value from the first data source available. This priority list is manageable by the user and it's different for each hazard, so that it may respect reasonable criteria of selection that could be related to availability and reliability of data sources. The complete set of values associated to an event of course shall not be lost and it's indeed available through detail views of single events.

Figure 2. Scatter chart of events



Source: Risk Data Hub, 2019

2.3 Presenting aggregated data for high level analysis

After implementing map viewers for spatial visualization and collecting a consistent amount of data for showcasing the analysis capabilities of RDH application, it started to become clear that something more was needed to make a sense of all data collected.

While map viewers allowed to perform in deep analysis of available datasets, the application was still missing a higher level output, something more about aggregations at country level of both Exposure (Risk) and Losses analysis and could better address the needs of a policy maker or simply someone looking for high level and easy comparable figures.

Thus started to take shape a new module called **Facts & Figures** which, as map viewers, is split in two sections: one about Exposure (it will become Risk once the layer of vulnerability will be developed) and one about Damages & Losses. This module, still presented as a prototype, is the main result of the efforts put during the current phase of development to improve the methodologies proposed by RDH.

Please, also note that when considering potential impacts, this document is always referring to Exposure, not Risk, as RDH has not included Vulnerability in its methodology yet.

We know that in Disaster Risk Management, Risk may be computed using the equation:

Risk = f(Hazard, Exposure, Vulnerability)

Missing the Vulnerability as a component of Risk, means that we are assessing only the Exposure to Hazards, which consists of the **maximum potential impact** in a given location.

What is discussed below it's just a partial extraction of what is visible on the website and it focuses on how data is shaped to give a particular output.

2.3.1 Exposure

The main goal here is to present how different hazards have an impact on various assets (or sectors) and to compare data of different countries, or any administrative unit data is available for.

The first thing to do to make data comparable is to transform absolute values into **relative values** (typically expressed in percentage); having this done, it's possible to represent which are the most exposed assets and what hazards this level of exposure comes from. Figure 3 shows an example of how assets are exposed to various hazards.

Figure 3. Incidence of hazards on specific assets

	Relative exposure to hazards		
	River floods	Coastal floods	Landslides
Commercial Built-up	9,5%	3,6%	0,8%
Residential Built-up	4,9%	2,5%	1,5%
Population	4,6%	2,5%	1,9%

This looks already good, but we can go a little bit further with our analysis, thinking of what we want to achieve when assessing Exposure (and finally Risk).

This is all about planning, so the desired output would be a **quantitative assessment** of severity and likelihood of impacts, for a given temporal horizon, which could be of 2 years, 5 years, or more.

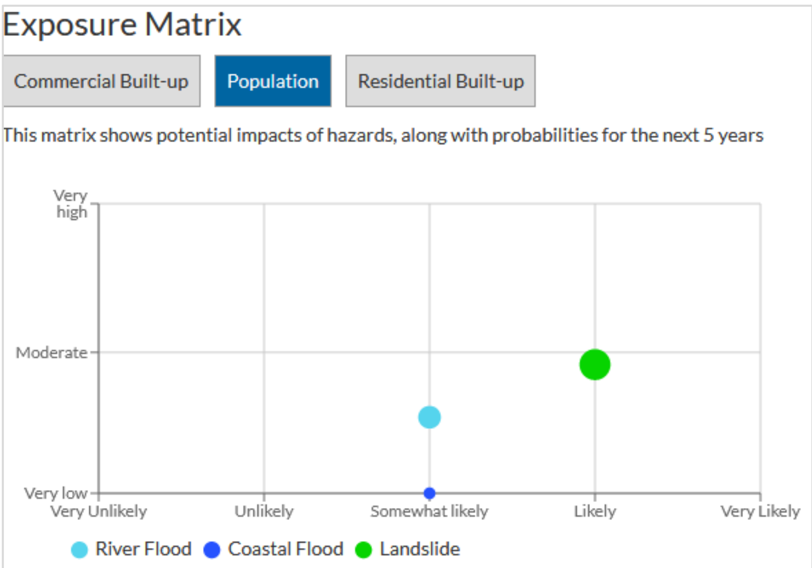
Mapping severity and likelihood of impacts would lead us to build a Risk Matrix, which in our case, we will call **Exposure Matrix**, as we're still missing Vulnerability, as previously reported. With this type of data visualization, we don't show absolute values, nor relative values: this is a qualitative representation, so for each element we want to introduce in the matrix (Hazards), we will assign a score like *Low*, *High*, or *Very High* for both severity and likelihood.

So, how do we get kind of result? First of all, we have to choose the dimension of the matrix: it could be 4x4, 5x5, 6x6 and so on. Since the most used one in literature appears to be 5x5, we'll be sticking on that format.

In the end, each value from our datasets has to be mapped to one of the 5 values accepted by the matrix, so we'll cluster data into 5 groups. Of course, **data clustering** needs to be done according to the elements we want to show in the matrix: for showing hazards, we'll fetch data for all hazards affecting one asset, or a reasonable set of homogeneous assets together, and take 5 groups out of them.

Figure 4 shows an example of Exposure Matrix of Hazards affecting Population.

Figure 4. Exposure matrix



Source: Risk Data Hub, 2019

Using Exposure Matrix let us compare hazards, assets potentially impacted and shows us how severity and likelihood may change in a different temporal horizon, which is very important, as a mean to assess the **cost of inaction**.

The methodology for projecting these figures to different temporal horizons is under development and the current version is presented below.

2.3.2 Methodology for constructing a Risk Matrix

An example case is presented at the end of this section and it shows the calculations performed for creating the matrix for exposure of commercial built-up to river floods, coastal floods and landslides in Europe for a certain time period.

In these calculations, the assumption is that the Exposed Area (E_T) is equivalent to the loss associated for each event. In the future a calculation of the associated loss will include a factor for the vulnerability.

The aim is to create a risk matrix illustrating the probability of occurrence for a disaster event on the one axis and the related expected loss (or impact) on the other axis for different hazards and different time-periods.

Depending on the hazard, the probability for certain events of different magnitudes needs to be defined for this method to work. Currently we only consider events from hazards for which we have specific return periods defined. The probability of exceedance is the number of times a stochastic process exceeds some critical value, in this case related to the return period, per unit time. This probability of exceedance is related to the probability of any single event to occur of a certain magnitude in the following way:

$$P_{T_n} = 1 - \prod_{i=T_1}^{T_n} (1 - p_i)$$

Where P_{T_n} is the probability of exceedance for an event with a return period of T_n and p_i the probability of occurrence for a single event. Solving this formula we get an expression for the probability of occurrence for an event with a return period T_n:

$$p_n = \frac{P_{T_n} - 1}{\prod_{i=T_1}^{T_{n-1}} (1 - p_i)} + 1$$

Where p_n is the probability of occurrence of an event with a return period T_n.

The expected loss for a given event is the probability multiplied by the exposed area and the vulnerability. Since we do have a value on the vulnerability, we set the vulnerability to one and assume that the value of the exposure is a representation of loss associated to the various magnitudes of events. The events are assumed to be independent, since at this point we are not able to establish any correlations between single events. Thus the overall average expected annual loss may be calculated by adding up the expected losses for all individual events in a given time period.

The overall average loss expected, U, for all events in *j* year/-s is simply:

$$U_j = \sum_{i=T_1}^{T_n} p_{i,j} L_i$$

Where L_i is the loss associated with for single event.

Since the starting point for all calculation in our case is the time period of 1 year, it is necessary to calculate the probabilities over longer time periods. This is done using the following formula:

$$p_T(j) = 1 - (1 - p_T)^j$$

Where j is the number of years.

Example 1: Commercial Built-up, River Floods – Coastal Floods - Landslides

The input table below shows values for exposure of Commercial built-up taken from the Risk Data Hub for River Floods for different return periods (data from the Risk Data Hub).

Table 1. Input table of areas exposed to River Floods for different return periods

Return Period, T (years)	***Probability of exceedance, P_T***	**River Floods: Exposed area, E_T (km2)**
500	*0.002*	720
200	*0.005*	670
100	*0.01*	628
50	*0.02*	581
10	*0.1*	434

For the event with the highest return period, i.e. T = 500 years, the exceedance probability is equal to the probability of occurrence. From that number it is possible to calculate all the individual probabilities associated to events with different return periods using the first two formulas in the methodology presented above.

$$P_{T_{500}} = p_{500}$$

$$P_{T_{200}} = 1 - (1 - p_{500})(1 - p_{200})$$
$$p_{200} = \frac{P_{T_{200}} - 1}{(1 - p_{500})} + 1$$

$$P_{T_{100}} = 1 - (1 - p_{500})(1 - p_{200})(1 - p_{100})$$
$$p_{100} = \frac{P_{100} - 1}{(1 - p_{500})(1 - p_{200})} + 1$$

$$P_{T_{50}} = 1 - (1 - p_{500})(1 - p_{200})(1 - p_{100})(1 - p_{50})$$
$$p_{50} = \frac{P_{50} - 1}{(1 - p_{500})(1 - p_{200})(1 - p_{100})} + 1$$

$$P_{T_{10}} = 1 - (1 - p_{500})(1 - p_{200})(1 - p_{100})(1 - p_{50})(1 - p_{10})$$
$$p_{10} = \frac{P_{10} - 1}{(1 - p_{500})(1 - p_{200})(1 - p_{100})(1 - p_{50})} + 1$$

Table 2. Probability of occurrence computed from probability of exceedance

Return Period, T (years)	***Probability of exceedance, P_T***	***Probability of occurrence, p_T***
500	*0.002*	0.002
200	*0.005*	0.003
100	*0.01*	0.005
50	*0.02*	0.010
10	*0.1*	0.082

Using the values calculated for 1 year it is now possible to move on and calculate the probabilities and overall average loss expected for different time periods: 2, 5, 10, 15 and 25 years. The first step is to calculate the probabilities of occurrence for each event over a selected time interval, n years:

$$p_T(n) = 1 - (1 - p_T)^n$$

$$n = 2, 5, 10, 15, 25\ [years]$$

$$T = 500, 200, 100, 50, 10\ [years]$$

Table 3. Probabilities of occurrence computed for different time periods

Return Period, T (years)	$p_T(1\ year)$	$p_T(2\ years)$	$p_T(5\ years)$	$p_T(10\ years)$	$p_T(15\ years)$	$p_T(25\ years)$
500	0.002	0.004	0.010	0.020	0.030	0.049
200	0.003	0.006	0.015	0.030	0.044	0.073
100	0.005	0.010	0.025	0.049	0.073	0.118
50	0.010	0.020	0.049	0.097	0.141	0.224
10	0.082	0.157	0.348	0.573	0.721	0.881

The overall average expected losses and probabilities of exceedance are then expressed simply by:

$$U_1 = p_{500,1}E_{500} + p_{200,1}E_{200} + p_{100,1}E_{100} + p_{50,1}E_{50} + p_{10,1}E_{10}$$
$$P_{T_{10,1}} = 1 - (1 - p_{500,1})(1 - p_{200,1})(1 - p_{100,1})(1 - p_{50,1})(1 - p_{10,1})$$

$$U_2 = p_{500,2}E_{500} + p_{200,2}E_{200} + p_{100,2}E_{100} + p_{50,2}E_{50} + p_{10,2}E_{10}$$
$$P_{T_{10,2}} = 1 - (1 - p_{500,2})(1 - p_{200,2})(1 - p_{100,2})(1 - p_{50,2})(1 - p_{10,2})$$

$$U_5 = p_{500,5}E_{500} + p_{200,5}E_{200} + p_{100,5}E_{100} + p_{50,5}E_{50} + p_{10,5}E_{10}$$
$$P_{T_{10,5}} = 1 - (1 - p_{500,5})(1 - p_{200,5})(1 - p_{100,5})(1 - p_{50,5})(1 - p_{10,5})$$

$$U_{10} = p_{500,10}E_{500} + p_{200,10}E_{200} + p_{100,10}E_{100} + p_{50,10}E_{50} + p_{10,10}E_{10}$$
$$P_{T_{10,10}} = 1 - (1 - p_{500,10})(1 - p_{200,10})(1 - p_{100,10})(1 - p_{50,10})(1 - p_{10,10})$$

$$U_{15} = p_{500,15}E_{500} + p_{200,15}E_{200} + p_{100,15}E_{100} + p_{50,15}E_{50} + p_{10,15}E_{10}$$
$$P_{T_{10,15}} = 1 - (1 - p_{500,15})(1 - p_{200,15})(1 - p_{100,15})(1 - p_{50,15})(1 - p_{10,15})$$

$$U_{25} = p_{500,25}E_{500} + p_{200,25}E_{200} + p_{100,25}E_{100} + p_{50,25}E_{50} + p_{10,25}E_{10}$$
$$P_{T_{10,25}} = 1 - (1 - p_{500,25})(1 - p_{200,25})(1 - p_{100,25})(1 - p_{50,25})(1 - p_{10,25})$$

The results from the calculations are shown in the table below.

Table 4. Average expected losses and exceedance probabilities from River Floods calculated for 6 time periods

	Overall average loss (U)	Probability of exceedance (P_T)
1 year	47.9	0.10
2 years	92.8	0.19
5 years	212.0	0.41
10 years	369.9	0.65
15 years	491.7	0.79
25 years	670.6	0.93

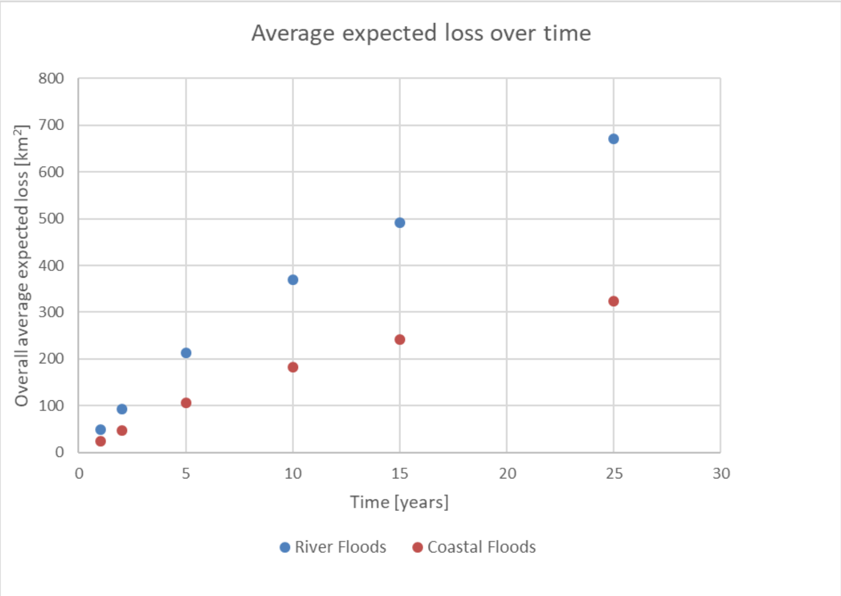
This same methodology can be used for Coastal Floods and Landslides and yields the following results.

Table 5. Average expected losses from Coastal Floods and Landslides calculated for 6 time periods

	Coastal floods		Landslides	
	Overall average loss (U)	Probability of exceedance (P_T)	Overall average loss (U)	Probability of exceedance (P_T)
1 year	24.0	0.10	6.3	0.50
2 years	46.5	0.19	11.8	0.75
5 years	105.7	0.41	26.0	0.97
10 years	182.7	0.65	45.0	1.00
15 years	240.8	0.79	60.5	1.00
25 years	323.2	0.93	85.0	1.00

Figure 10 simply illustrates on a chart the result of these calculations, excluding Landslides only for visualization purposes, as its values are much lower than other hazards.

Figure 5. Average expected losses from River and Coastal Floods over time



Source: Risk Data Hub, 2019

The above chart reflects what we could already guess even without any particular notion of probability calculation: the average expected loss is increasing over time.

Now that we have average expected loss, in order to be able to compare different hazards, as explained above in chapter 2.3.1, we need to select one asset (e.g. Commercial Built-up space) and use relative values instead of absolute values.

The likelihood of having some losses is increasing as well over time, as shown in figure 5; for simplicity, this probability is referred to the type of event that causes the minimum loss, so this value could also be interpreted as the probability that any loss happens.

In the end, constructing the Risk Matrix is a matter of clustering values of both expected loss and probability into *n* groups, according to the dimensions of the desired matrix.

2.3.3 Damages & Losses

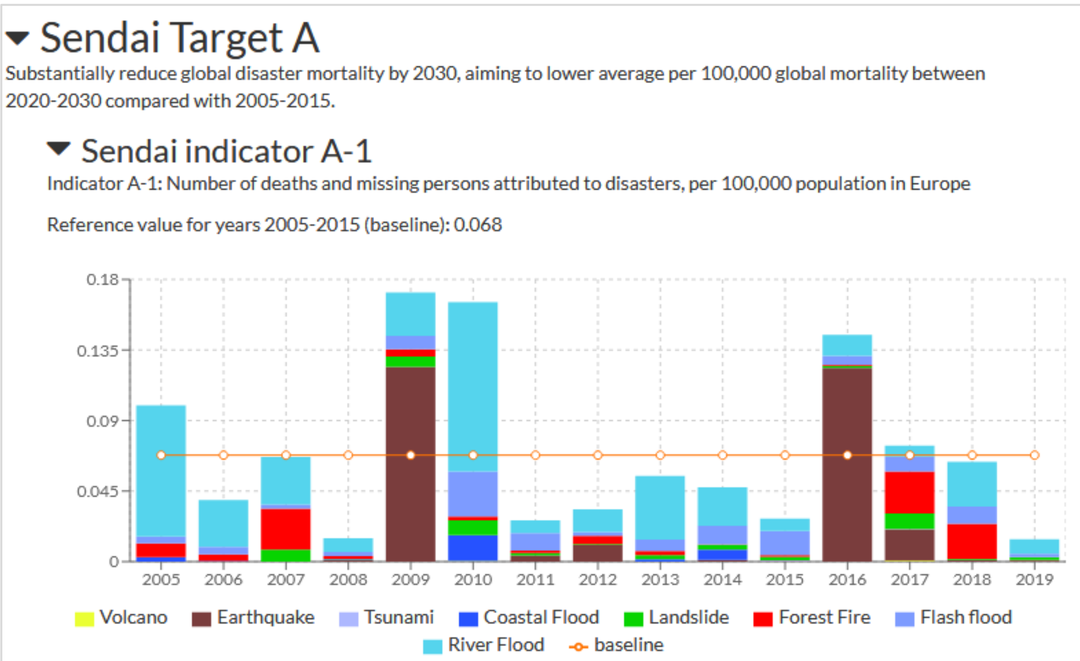
The approach followed by RDH to present data on damages and losses from past events was to structure information and prepare outputs compliant to Sendai Framework for Disaster Risk Reduction.

Explaining the motive behind this choice is out of the scope of this document, so this chapter focuses only on how data is handled and presented on the web platform. For details, please refer to the publication *Risk Data Hub – web platform to facilitate management of disaster risks, Antofie, T., Luoni, S., Faiella, A., Marin Ferrer, M. - JRC 2019*.

It is worth mentioning that while individual countries are responsible for reporting to Sendai, RDH uses data publicly available from all European countries to present an aggregated view at the continent level. This is particularly interesting, considering that in many countries the number of recorded events is rather limited, hence the difficulty of extracting significant information.

Going with the idea of having an output useful for reporting to Sendai, data is grouped by Sendai Target and Sendai Indicator; figure 6 shows an example of compound indicator A-1, where each bar denotes a year and is the result of adding up values of different hazards.

Figure 6. Bar chart for Sendai Indicator A-1



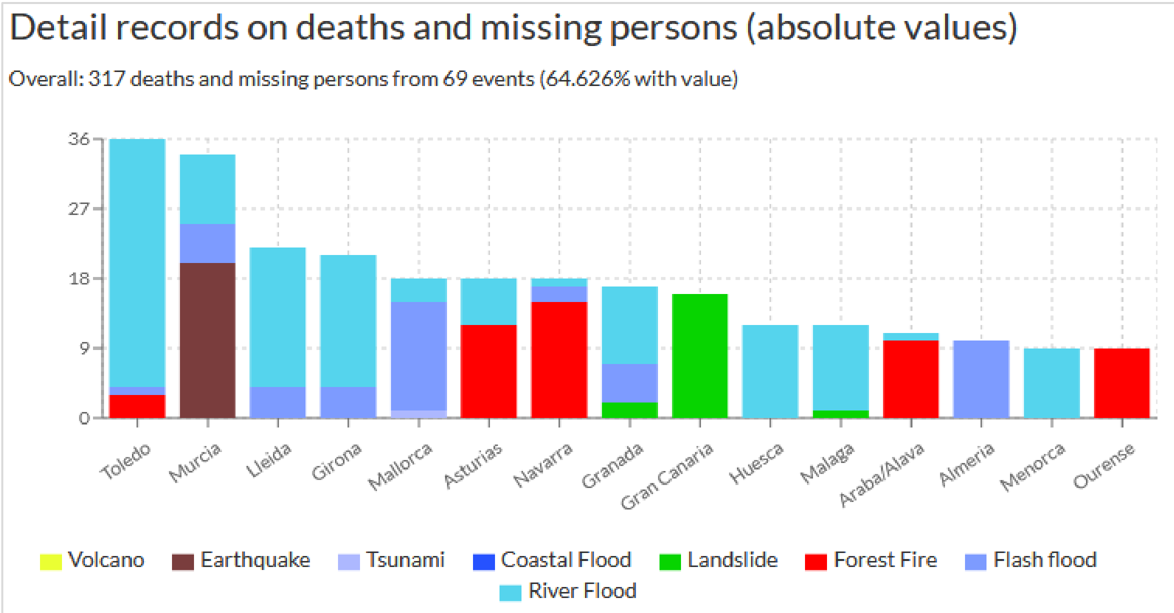
Source: Risk Data Hub, 2019

This particular representation comes with a technical problem, related to the definition of compound indicator and the availability of data. A compound indicator practically is the highest level score related to a certain Target, so for example Indicator A-1 collects data on deaths and missing persons and it's the general indicator for Target A which aims to reduce global disaster mortality. Indicator A-1 may then be disaggregated, so there are Indicator A-2 reporting only number of deaths and Indicator A-3 reporting only missing persons. It's clear that A-1 should be the sum of A-2 and A-3, but what happens many times in reality is that data available is limited to compound indicator; in a case like this, we simply don't have any values for A-2 and A-3, so we would need to show values for A-1 as we get them, because there is no sum operation to do.

This is still simple, but what to do if we manage to get data for a compound indicator and then some data for a single sub-indicator? This is particularly unfortunate, because it's a mixed situation where the first thing to do would be to check data sources to figure out whether the compound may already be including the data received for the other sub-indicator or not. Here we suppose that there is no overlapping, so the sub-indicator should be used to calculate the compound: RDH is able to handle this and will re-calculate the compound from sub-indicators, even if not all of them are filled with data.

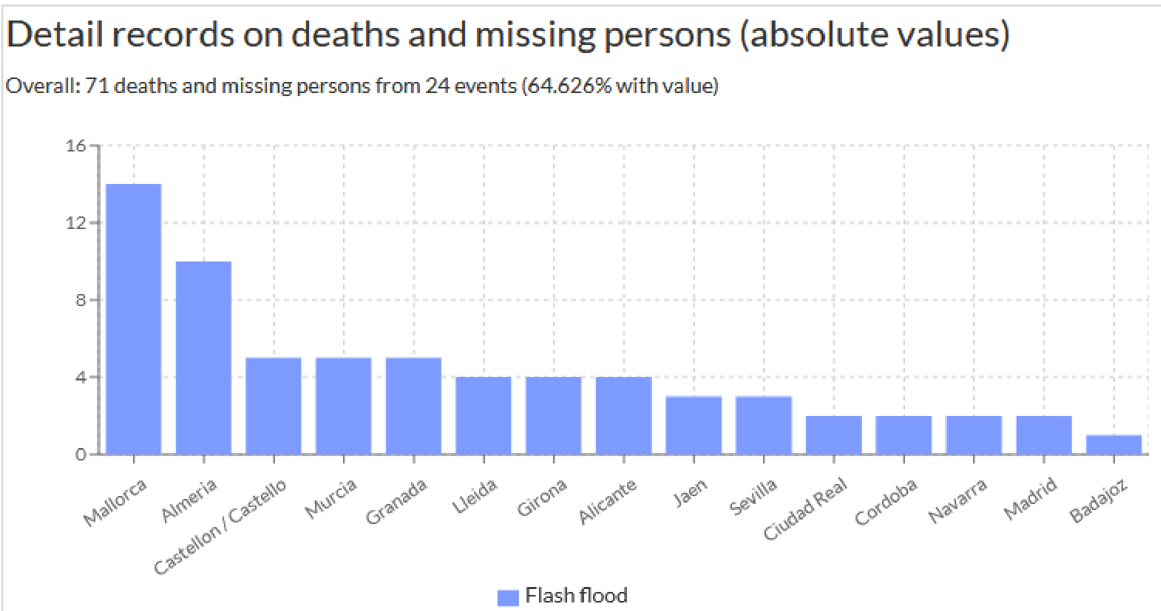
After these considerations on data availability, we want to know how to get something more out of D&L data: RDH, within Facts & Figures module, currently proposes to **filter by Hazard, Year and Location**. Playing with these filters is a simple way to watch data from different angles, identify most critical locations and trends. Figure 7 and 8 show examples of data on deaths and missing persons in Spain, with no filters and filtered by Hazard (Flash Flood).

Figure 7. Total number of deaths and missing persons in Spain from 2005 to 2019



Source: Risk Data Hub, 2019

Figure 8. Total number of deaths and missing persons in Spain from 2005 to 2019, filtered by Flash Flood



Source: Risk Data Hub, 2019

2.4 Refactoring data access layer for improved performance

This sub-chapter describes backend development activities, which didn't have any visibility on the web interface, as they do not reflect to any features, but they represent a crucial part of the architecture currently adopted by RDH.

In its first version, RDH was tightly coupled with Geonode and GeoServer: all layers were uploaded through the Geonode interface and then rendered on maps after calling WMS services exposed by GeoServer. This is a standard approach for web GIS applications, but it wasn't the best possible practice, considering datasets used in RDH.

Even if the platform may be used to just create maps using multiple layers (from both shapefiles or rasters), the key point with RDH is not to render on maps some raw data as it is, but to follow a methodology that requires organizing data and gives a structured output that makes sense of them. Most of data stored by RDH are tabular data linked to administrative units (e.g. NUTS2, NUTS3 and LAU), so they're geo-referenced in a way, but they don't need to include (and thus replicate) any geometries, as polygons for any administrative units would be introduced in the database in advance and only once.

Having said this, in the first version of RDH, all main datasets were already created after tabular data, but they were exposed to the frontend of the application only through GeoServer, which means that they were extracted calling a layer defined in GeoServer by a SQL view, executed on its data store. This was a good way for a quick implementation that would expose all data using OGC compliant WMS and WFS services, but had three critical downsides:

- The use of SQL queries into GeoServer implied having logic outside of the codebase of the application; this was a bad practice, as using and ORM is normally the preferred way to access data from models
- The Geo Web Cache (GWC) implemented by GeoServer couldn't work with dynamic SQL views, so this also implied a waste of computational resources
- As data layers were defined as SQL views into GeoServer, also the style for these layers needed to be defined into GeoServer, using SLD standards

A second phase of development was then addressed to refactor all the code involved in data extraction. The results could be summarized by the point below.

1. Data extraction using Django ORM: no more logic into GeoServer
2. No-SQL solution to index data for faster data extraction: datasets were denormalized using Postgres JSONB data type
3. Automatic generation of styles for datasets, based on the same data clustering used for Exposure Matrix.
4. Use of REDIS as backend cache

The impact of points listed above is clearly significant, but it is actually greater than what it seems at a first sight. Regarding data extraction, is has been explained that everything, no matter it was a shapefile, a raster file, or tabular data to be linked to the geometry of administrative units, was saved as a layer by Geonode and retrieved, for data visualization, by calling GeoServer services. As the communication with GeoServer was implemented by Geonode library, removing GeoServer from the data extraction chain (at least for RDH main datasets) leads to a loose coupling between RDH and Geonode, which is a great advantage for both development and maintenance endeavours.

Furthermore, using only one vector layer on maps, with styles applied by the client, instead of retrieving tiles already coloured by GeoServer, allows reducing a lot the overhead, as the application is able to dynamically change styles (e.g. after applying filters) without the need to re-extract data.

3 Technologies used

This chapter is about technology selection and architectural design of RDH application.

3.1 Overall architecture

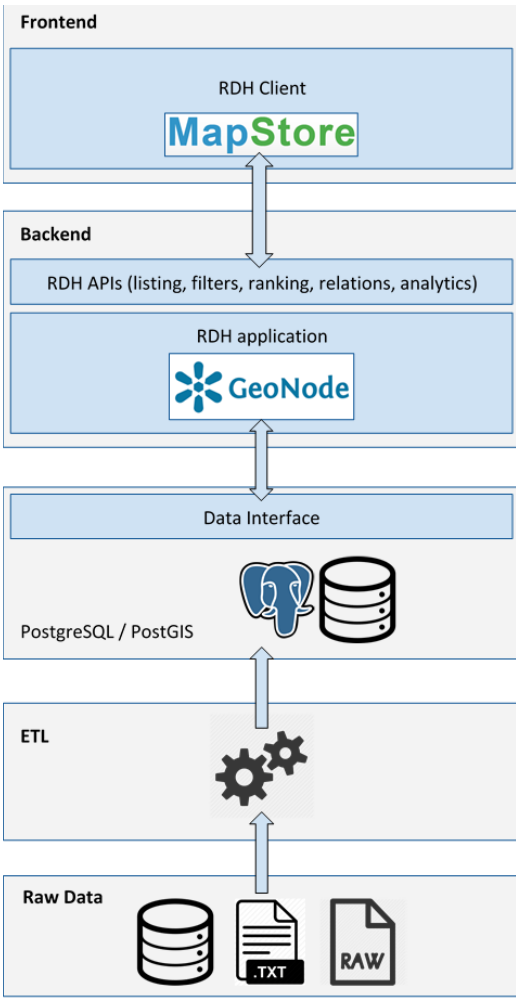
The second phase of the development of Risk Data Hub application ended with some important updates, as anticipated by previous chapter. From an architectural perspective, the most significant change was about moving from a tight to a loose coupling between RDH and Geonode.

Another part of the solution implemented that is worth of a more detailed explanation is related to data models and consists of the usage of Postgres JSONB data types (for details, see chapter 4).

The system architecture as a whole is quite articulate and makes use of several tools to perform all operations needed. Basically, the project is built with Django (Python web framework), using Geonode as dependency, PostGIS as database backend and a client application developed with ReactJS.

Figure 9 illustrates a simple schema of the architecture implemented for RDH, which is actually unchanged from the last report on software architecture, as the main software and components used by the different layers in architectural scheme are still the same.

Figure 9. RDH software architecture



Source: Risk Data Hub, 2019

Let's a have a more detailed look on the single pieces of architecture.

3.2 Data harvesting and ETL

Data is harvested from multiple heterogeneous sources and loaded into RDH database by ad hoc Python scripts. Relevant operations involved by data ingestion process are:

- Definition and scheduling of importing jobs
- DBs health check
- Grouping data into multiple layers
- Pre-calculate relevant statistics
- Data clustering
- Normalize taxonomies
- Check and cast geometry fields
- Create style for different types of layers and geometries
- Import GeoServer layers in Geonode
- Populate keywords and categories from DB view attributes
- Populate title and description fields
- Define Geofence rules

3.3 Data Interface

The basic operations performed by RDH application against PostGIS database are:

- Data extraction and pre-processing (Django ORM + RDH custom logic)
- Spatial queries to extract spatial relations between datasets
- Extract administrative division boundaries

The basic operations performed by RDH application against GeoServer are:

- OGC/WMS service calls to view layers on map (only overlays introduced from Geonode interface)
- Geofence rules to restrict access to layers and services
- GeoWebCache for tile caching

3.4 Backend

The core of RDH application was developed in Python (Django framework). The backend code includes the definition of models and relevant data access layer, various utility scripts and RESTful APIs.

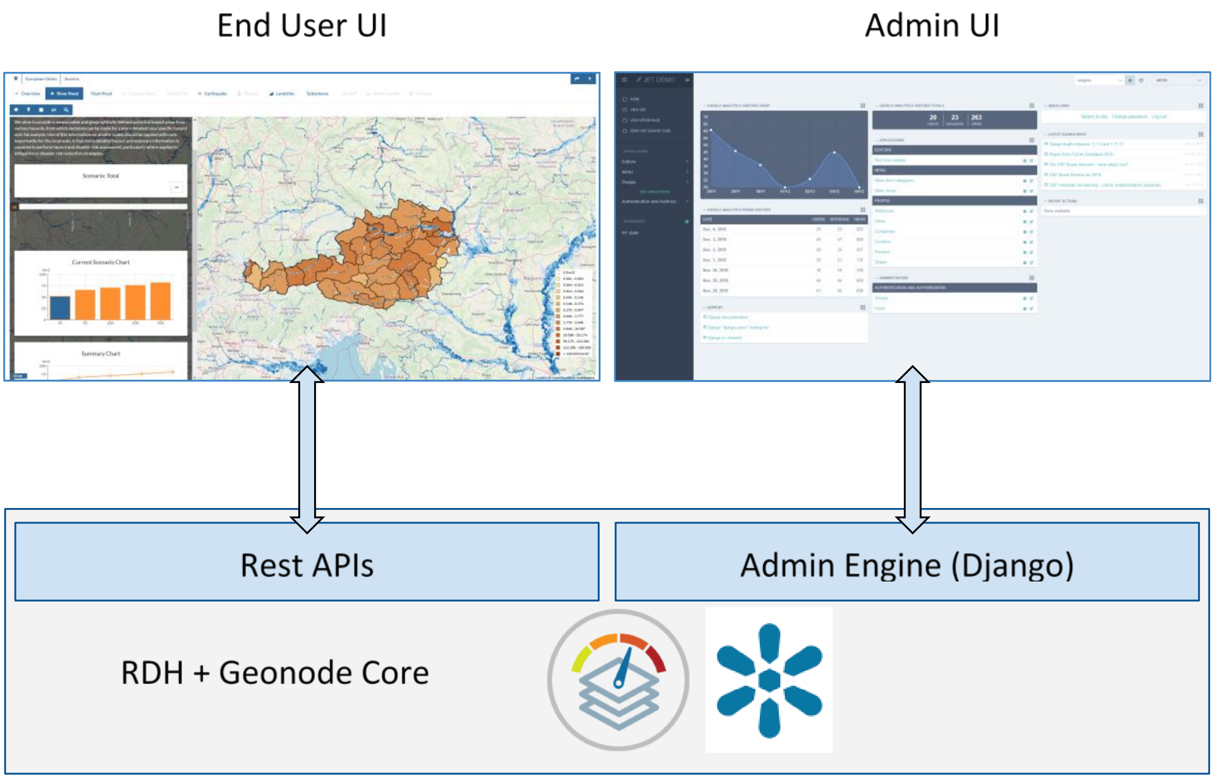
Geonode is used for uploading and managing complementary spatial data (shapefiles or raster files) that are included in maps as overlays. Its models and APIs are used as well for:

- Enrich original data with metadata and additional informations (keywords and categories)
- Support frontend functionalities
- Publish a CSW catalogue of the layers
- Consume Geoserver APIs for management commands
- Proxy WMS requests under ACLs

3.5 Frontend

The frontend is based on Mapstore framework for web mapping and it uses some of its core components along with custom components to build the User Interface. It is a single page application developed with React JS and Leaflet maps.

Figure 10. Interaction of User Interfaces with backend



Source: Risk Data Hub, 2019

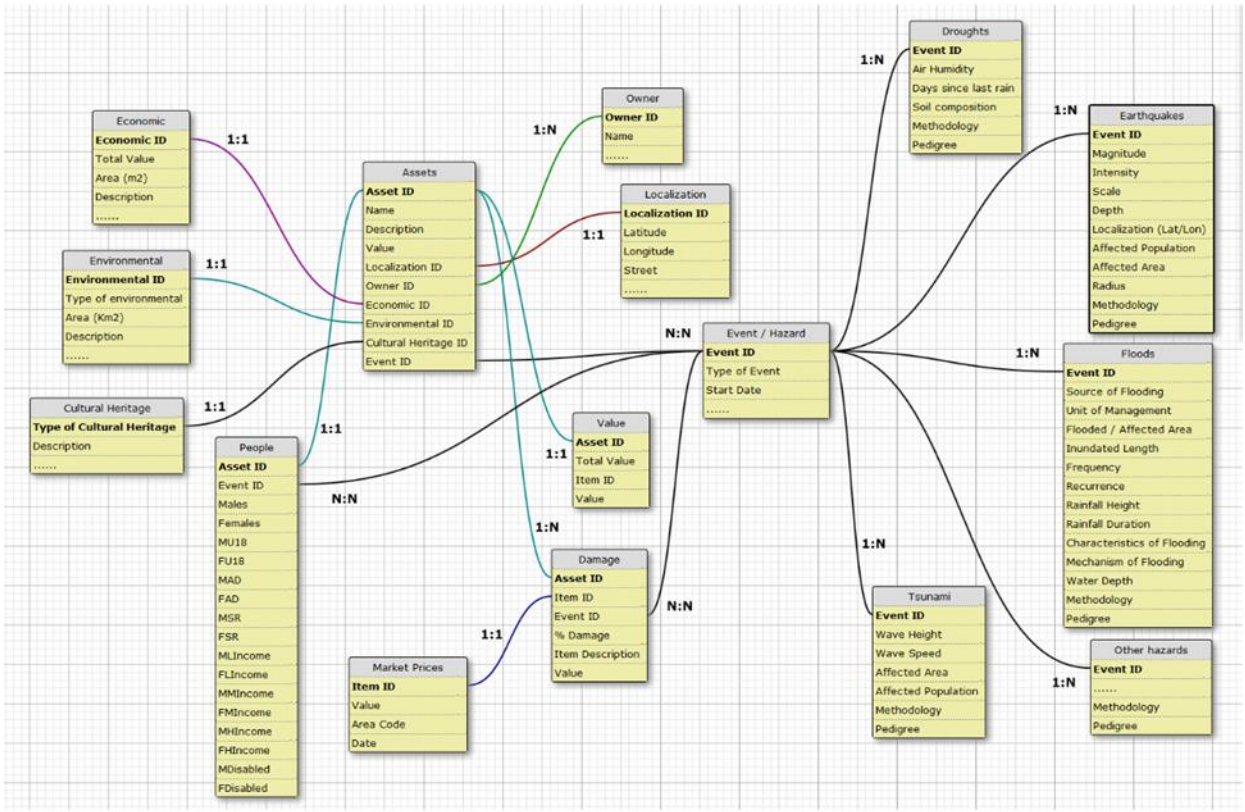
4 Database architecture

4.1 Evolution of Loss Database for Disaster Risk Management

The Loss Database for Disaster Risk Management was initially designed after a process of analysis of several databases, developed following different purposes for data collection and most of the times, focused on one single hazard.

The idea was to propose “a common structure of a generic database able to accommodate and properly record the required particularities of a vast variety of events triggered by any kind of hazard” (JRC, 2018).

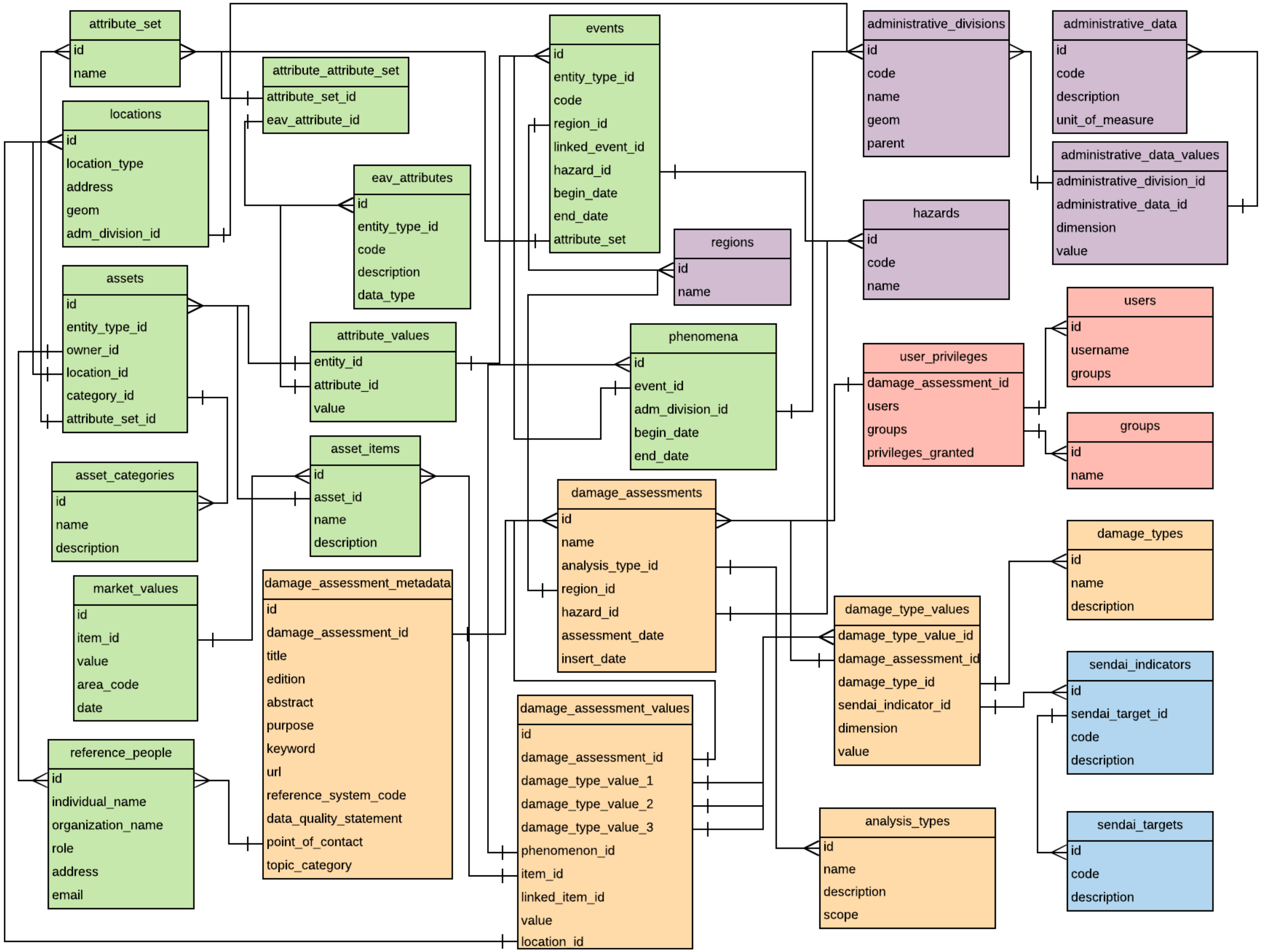
Figure 11. Loss Database diagram as of EUR 29063 EN publication



Source: Risk Data Hub, 2019

That principle of a comprehensive data model was adopted and furtherly developed by Risk Data Hub. Figure 12 shows the database diagram used for RDH, after the first part of the development.

Figure 12. RDH Database diagram as of EUR 29756 EN publication



Please, note that in the image above, tables are highlighted in different colours, corresponding to specific functionalities within the application.

For references, see *Risk Data Hub software and data architecture, JRC 2019 - EUR 29756 EN* and *Update of the Disaster Risk Management Knowledge Centre loss database architecture for disaster risk management, JRC 2019 – EUR 29063 EN* publications.

What has changed?

4.1.1 Data Source

The management of multiple sources for data, applied with the concept of priority list described in chapter 2.2, requires defining a new Data Source entity, which has to be related to the Hazard entity, as we want to have independent settings per hazard.

4.1.2 Phenomenon

Phenomenon entity has changed to include latitude and longitude as part of its key and also a field for storing geometry, to make possible what described in chapter 2.1.

4.1.3 Location

The Location entity was removed, as Phenomenon has become the entity of choice for storing all spatial information for damages and it has been considered that there was no need for Asset entity to be linked to a separate Location entity.

4.1.4 Access Rule

Access rules are used to manage permissions on datasets. This is not a relevant part of the logical schema, as this entity only stores some settings used to check whether a request for data should be authorized or not. Access Rules may be used restrict access of some data to a particular user, or a group of users. Data affected by these rules may be everything is owned by the user managing this setting, or a single dataset, or even only some data within a dataset: for instance a user could decide to grant public access to aggregated values at the level of the country, but restrict access to values related to lower administrative units.

4.1.5 Region

Region entity, which used to be referenced by instances of Event and Damage Assessment (now called Assessment), is now referenced also by Access Rules, Administrative Data, Data Source and Asset. This particular entity is used to separate data and settings for each User Corner.

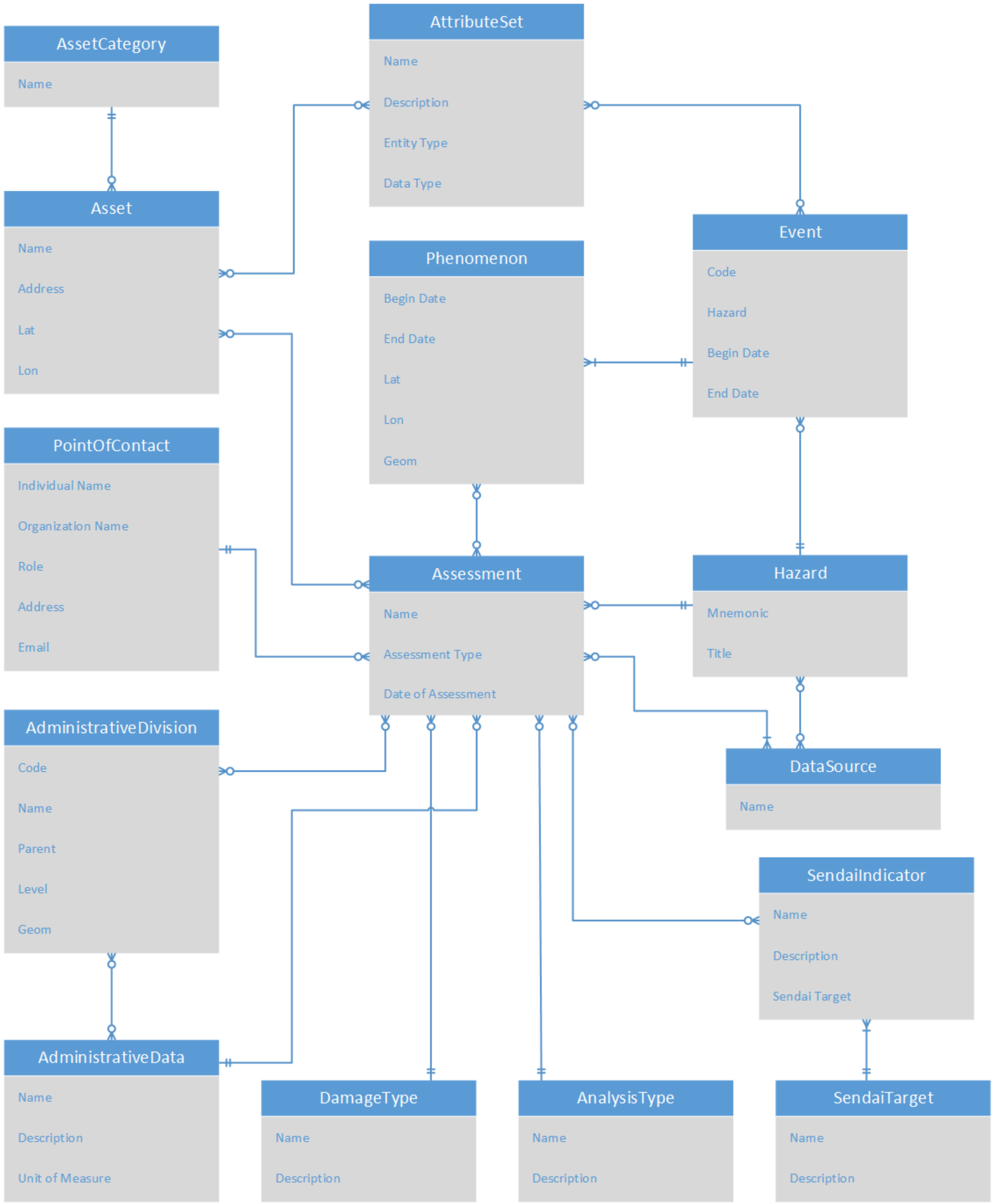
4.1.6 Assessment Entry

This was the major change in the database structure. Basically, we still have an entity to store basic information of the datasets (Assessment); then we have entities for storing exposure/risk data (Entry Values) and losses data (Entry Values Event). All data in the end is denormalized and stored as JSON objects, using the JSONB data type available from Postgres database.

4.2 New RDH database

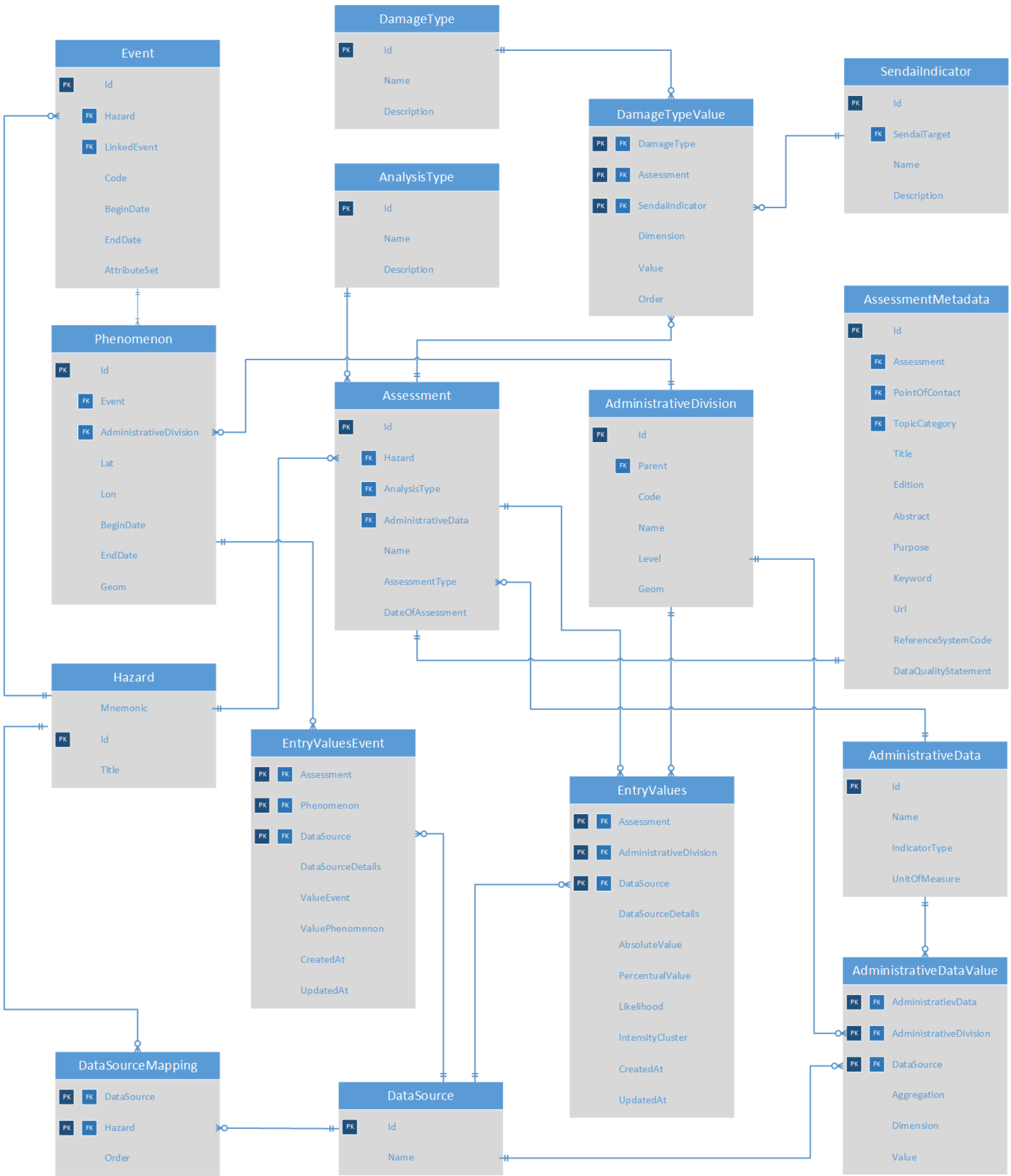
Figure x and y show a logical and a pseudo-physical diagram of the database. Please, note that figure y actually is not a complete representation of physical diagram, as data types have been omitted for saving space. Some entities have been omitted as well, in order to have a more compact picture and focus on main tables used for storing assessment data.

Figure 13. Database logical diagram



Source: Risk Data Hub, 2019

Figure 14. Database pseudo-physical diagram

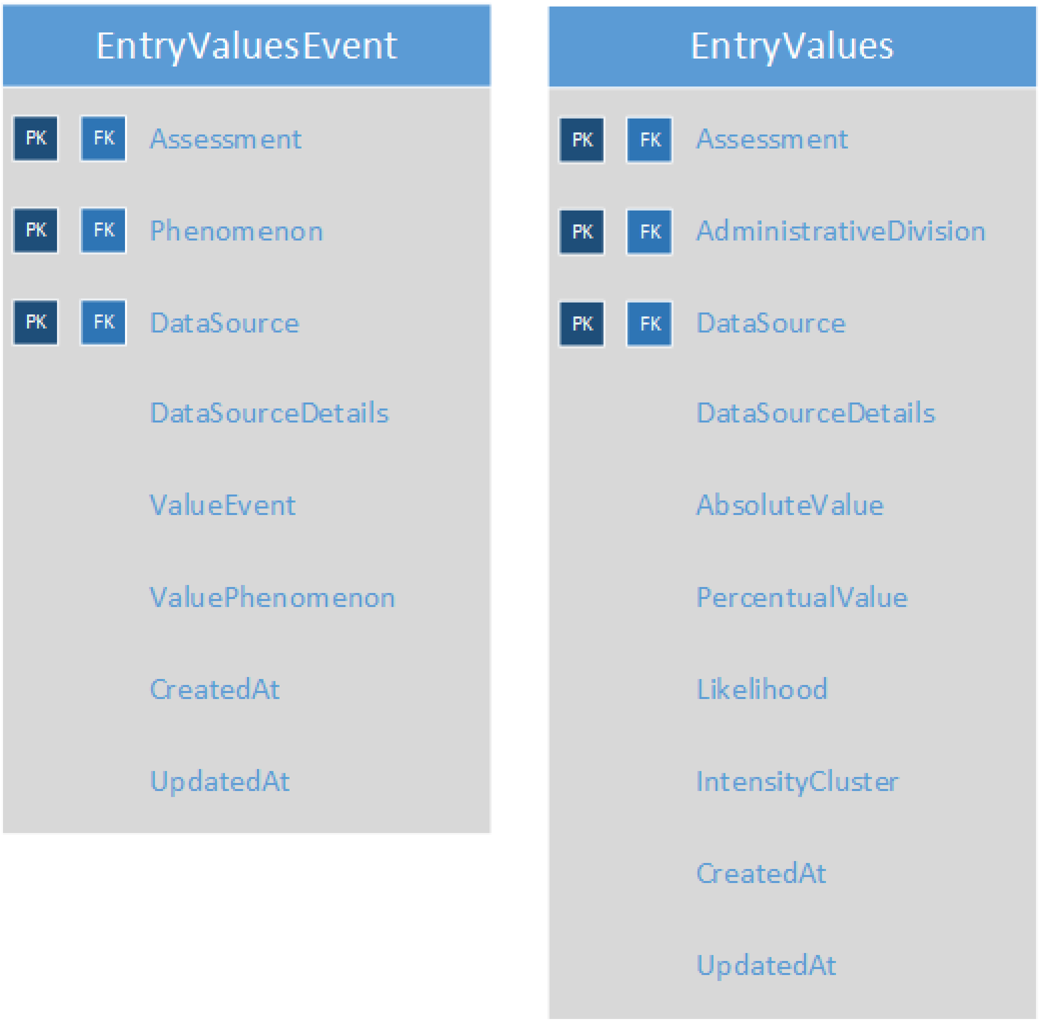


Source: Risk Data Hub, 2019

EntryValues and EntryValuesEvent contain all fields needed to store data for Exposure/Risk datasets and Damages&Losses datasets. As anticipated in previous chapters, all data is finally indexed in JSON objects, adopting a No-SQL solution for speeding up data extraction. The RDH backend code defines abstract classes for EntryValues and EntryValuesEvent, so what happens in reality is that these models do not exist physically in the database, but are only used for serializing JSON objects. That’s why the diagram in figure 13 is called pseudo-physical: it’s not a physical diagram indeed, but it clarifies what models are used, going to a lower level than the logical diagram.

Figure 15 shows an example of a JSON object for a damages&losses dataset.

Figure 15. Data tables



Source: Risk Data Hub, 2019

Figure 16. Example of Assessment entry in JSON format


```
{
  "dim1": {
    "id": 293,
    "axis": "x",
    "name": "Scenario",
    "unit": "NA",
    "order": 0,
    "value": "burned_area",
    "layer_attribute": "dim1"
  },
  "dim2": {
    "id": 294,
    "axis": "y",
    "name": "Return Period",
    "unit": "Years",
    "order": 0,
    "value": "Event_value",
    "layer_attribute": "dim2"
  },
  "event": {
    "id": 3680,
    "code": "FFES201709010044",
    "year": 2017,
    "state": "ready",
    "hazard": "Forest Fire",
    "country": "ES",
    "end_date": "2017-11-30",
    "begin_date": "2017-09-01"
  },
  "hazard": "FF",
  "region": "Europe",
  "values": [
    {
      "data_source": "EFFIS",
      "insert_date": "2019-07-24",
      "last_update": "2019-07-24",
      "value_event": 10,
      "phenomenon_id": 35659,
      "value_phenomenon": null,
      "data_source_details": "EFFIS_ID: 32856"
    }
  ],
  "phenomenon": {
    "id": 35659,
    "end_date": "2017-10-15",
    "event_id": 3680,
    "begin_date": "2017-10-15"
  },
  "damage_assessment": {
    "name": "FF_burned_area",
    "unitOfMeasure": "ha",
    "relatedAdministrativeData": "Area"
  },
  "administrative_division": {
    "id": 1017,
    "code": "ES423",
    "name": "Cuenca",
    "level": 3,
    "hierarchy": [
      "EU",
      "ES",
      "ES42",
      "ES423"
    ],
    "parent_id": 165
  }
}
```


Source: Risk Data Hub, 2019

4.2.1 List of tables

Asset

Description: generic entity affected by event (includes also People)

Fields:

- Id (int): unique identifier
- Entity_type (enum): defines entity type for mapping fitting attributes
- Owner_id (int): reference to PointOfContact
- Category_id (int): reference to AssetCategory
- Address (varchar 100)
- Country (varchar 100)
- Lat (decimal)
- Lon (decimal)
- Attribute_set_id (int): reference to AttributeSet
- Details (jsonb): attributes in attribute set are stored in json format

AssetItem

Description: single item included in the asset (equals to asset in the simplest case)

Fields:

- Id (int): unique identifier
- Asset_id (int): reference to assets
- Name (varchar):

AssetCategory

Description: categories for assets; e.g. Buildings, Infrastructures or People

Fields:

- Id (int): unique identifier
- Name (varchar):
- Description (varchar):

MarketValue

Description: market value of items

Fields:

- Id (int): unique identifier
- Item_id (int): reference to assets
- Value (decimal):
- Area_code (varchar):

- Date (datetime): start validity date

PointOfContact

Description: could be the owner of an asset, author of publications, etc.

Fields:

- Id (int): unique identifier
- Individual_name (varchar):
- Organization_name (varchar):
- Role (varchar):
- Address (varchar):
- City (varchar):
- Zipcode (varchar):
- Country (varchar):
- Email (varchar):

EavAttribute

Description: attributes relevant to events and assets (and more) are defined in a single place. This feature allows to define new attributes at any time, without the need to change the structure of database.

Fields:

- Id (int): unique identifier
- Entity_type_id (int): defines entity type for mapping fitting attributes
- Data_type (varchar): defines data type (varchar, text, integer, decimal, datetime)
- Name (varchar):
- Description (varchar):

AttributeSet

Description: attribute sets are used to link attributes to specific instances of an entity

Fields:

- Id (int): unique identifier
- Name (varchar):

AttributeSetAttributes

Description: this is a relation between AttributeSet and EavAttribute, so it's basically the content of an attribute set

Fields:

- Attribute_set_id: reference to AttributeSet
- Eav_attribute_id: reference to EavAttribute

Event

Description: an event is a generic entity which may be the cause of a damage.

Fields:

- Id (int): unique identifier of the event
- Entity_type_id (int): defines entity type for mapping fitting attributes
- Region_id (int): could be Europe, or any country corner
- LinkedEvent_id (int): optional link to an event identified as cause of the current one (chained events)
- Hazard_id (int): identifier of the hazard (eg. Flood)
- Begin_date (datetime): starting date of recognized event
- End_date (datetime): starting date of recognized event
- AttributeSet_id: reference to AttributeSet

Phenomenon

Description: a phenomenon is part of a major event and has specific location and related assessed damage.

Fields:

- Id (int): unique identifier
- Event_id (int): related event
- AdministrativeDivision_id (int): reference to AdministrativeDivision
- Begin_date (datetime): starting date of recognized event
- End_date (datetime): starting date of recognized event
- Lat (decimal): latitude
- Lon (decimal): longitude
- Geom (MultiPolygon): the geometry associated to phenomenon

Hazard

Description: definition of Hazard (e.g. River Flood)

Fields:

- Id (int): unique identifier
- Mnemonic (varchar): e.g. FL for Flood
- Description (varchar):

DataSource

Description: data source (or provider, e.g. Eurostat)

Fields:

- Id (int): unique identifier
- Name (varchar)
- Region_id (int): reference to Region

- Colour (varchar): colour assigned (HEX code) to be used for legends

DataSourceMappings

Description: data source relations with Hazards

Fields:

- DataSource_id (int):
- Hazard_id (int)
- Order (int): used for priority list (see chapter 2.2)

AdministrativeDivision

Description: This entity stores basic data of administrative divisions.

Fields:

- Id (int): unique identifier
- Code (varchar): ISO2 for countries, or relevant NUTS code according to Eurostat
- Name (varchar): name of administrative division
- Geom (Multipolygon): spatial data
- Level (int): level of instance in the hierarchy (e.g. Continent = 0, Country = 1)
- Parent_id (int): parent adm division

Region

Description: this is crucial for ownership management of data and visibility. Each user in the system belongs to a specific Region and so are the data owned by that user.

Fields:

- Id (int): unique identifier
- Name (varchar): name of Region (e.g. Europe, or country corner, like Austria)

AdministrativeData

Description: definition of data related to Administrative Divisions, like GDP, Population, Area, and so on.

Fields:

- Id (int): unique identifier
- Code (varchar): e.g. GDP
- Description (varchar): description of data
- UnitOfMeasure (varchar): e.g. Mln EUR

AdministrativeDataValue

Description: relation between Administrative Data and Administrative Divisions.

Fields:

- Administrative_division_id (int):
- Administrative_data_id (int):
- Dimension (varchar): e.g. Year 2018 of GDP
- Value (decimal):

AnalysisType

Description: defines the type of data analysed (e.g. Polulation, Buildings, Econonic values)

Fields:

- Id (int): unique identifier
- Name (varchar):
- Description (varchar):

Assessment

Description: definition of data measured

Fields:

- Id (int): unique identifier
- Name (varchar): name given (unique)
- AnalysisType_id (int): reference to AnalysisType
- Region_id (int): reference to Region, needed for Risk Analysis that do not use events
- Hazard_id (int): reference to Hazard, needed for Risk Analysis that do not use events
- Scope (enum): identifies the type of assessment (exposure, or damages&losses)
- AdministrativeData_id (int): reference to AdministrativeData
- Assessment_date (datetime): date declared for the assessment
- Insert_date (datetime): date of insertion in the database

DamageType

Description: definition of considered scenario. It is useful for complex analysis with predicted values in different declinations of a given scenario (e.g. climate change)

Fields:

- Id (int): unique identifier
- Name (varchar): name given (unique)
- Description (varchar):

DamageTypeValue

Description: relation between Assessment and Damage_Type

Fields:

- Id (int): unique identifier
- Damage_assessment_id (int): reference to Assessment

- Damage_type_id (int): reference to DamageType
- Sendai_indicator_id (int): reference to SendaiIndicator
- Dimension (varchar): e.g. Axis of a chart
- Order (int): order of item in a sequence
- Value (varchar): value of damage type for given assessment and dimension

AssessmentEntry

Description: value assigned to the loss for given phenomenon, damage assessment, damage type and item

Fields:

- Id (int): unique identifier
- Assessment_id (int): reference to Assessment
- Phenomenon_id (int): reference to Phenomenon
- Entry (jsonb): content in json format (see Figure 15)

AssessmentMetadata

Description: complementary description of a damage assessment publication

Fields:

- Id (int): unique identifier
- Assessment_id (int): reference to Assessment
- Title (varchar):
- Edition (varchar):
- Abstract (varchar):
- Purpose (varchar):
- Keyword (varchar):
- Url (varchar):
- Reference_system_code (varchar):
- Data_quality_statement (text):
- Point_of_contact (int): point of contact for the publication (reference_people)
- Author (int): author of publication (reference_people)
- TopicCategory: e.g. Environmental, Structure, etc.

SendaiTarget

Description: Sendai Target as defined by UNISDR specifications

Fields:

- Id (int): unique identifier
- Name (varchar): (unique)
- Description (varchar):

SendaiIndicator

Description: Sendai Indicator as defined by UNISDR specifications

Fields:

- Id (int): unique identifier
- SendaiTarget_id (int): reference to target
- Name (varchar): (unique)
- Description (varchar):

User

Description: user registered to the site. Every user is assigned to a region.

Fields:

- Id (int): unique identifier
- Username (varchar): (unique)
- Region_id (int): reference to Region

Group

Description: group of users for permission purposes.

Fields:

- Id (int): unique identifier
- Name (varchar): (unique)
- Permissions (int): reference to Permission (see Django framework documentation)

AccessRule

Description: rule for filtering access to datasets. Users may define multiple rules, which will be applied according to the value of "Order" field.

Fields:

- Id (int): unique identifier
- Region_id (int): reference to Region
- Scope (enum): defines the type of assessment (see definition of table Assessment)
- Assessment_id (int): reference to Assessment
- Group_id (int): reference to group
- User_id (int): reference to User
- MinAdmLevel (int): minimum level for AdministrativeDivision to be accessible
- MaxAdmLevel (int): maximum level for AdministrativeDivision to be accessible
- AdministrativeDivisionRegex (varchar): regular expression to match for filtering Administrative Divisions
- Access (enum): defines type of rule (ACCESS or DENY)
- Order (int): used for determining order of execution of rules

Style

Description: style assigned to a dataset (automatically generated)

Fields:

- Id (int): unique identifier
- Assessment (int): reference to Assessment
- ParentAdministrativeDivision (int): reference to AdministrativeDivision
- Level (int): level of AdministrativeDivision style is applied for
- Content (jsonb): style in json format to be used in Leaflet maps

5 Conclusions

This report has presented development activities and updates of the Risk Data Hub platform during 2019, trying to use a descriptive approach, as the purpose was not to create a developer guide.

Most of the efforts have been addressed to the development of Facts & Figures module, which is for sure the biggest new feature introduced and will still need much work, either to validate methodologies, improving the user experience and the output produced.

The Risk Data Hub application as a whole is still on a development phase and different collaborations have been established with both scientists and end users from several areas. Future work will then focus on new topics, like technological disasters, critical infrastructures and cultural heritage; this is why the identification of new features to be integrated is expected.

The database architecture is by design hopefully flexible enough to handle all the complexity introduced, but as the application continues to grow, we cannot exclude that upgrades will be needed also for this part, which is typically the most critical to change.

References

Loss Database Architecture for Disaster Risk Management

(https://ec.europa.eu/jrc/en/publication/eur-scientific-and-technical-research-reports/loss-database-architecture-disaster-risk-management)

Ríos Díaz, F., Marín Ferrer, Montserrat: 2018 – Update of the DRMKC Loss database architecture for disaster risk management

Antofie, Tiberiu-Eugen, Luoni, S., Faiella, A., Marín Ferrer, Montserrat: Risk Data Hub – web platform to facilitate management of disaster risks

Entity-attribute-value model (EAV) – Wikipedia (https://en.wikipedia.org/wiki/Entity-attribute-value_model)

Extract Transform Load (ETL) – Wikipedia (https://en.wikipedia.org/wiki/Extract,_transform,_load)

CIMA Research Foundation (http://www.cimafoundation.org/)

EU Floods Directive (https://ec.europa.eu/environment/water/flood_risk/)

DRMKC Support Service System (https://drmkc.jrc.ec.europa.eu/innovation/SupportSystem)

Python programming language (https://www.python.org/)

Django (https://www.djangoproject.com/)

PostgreSQL (https://www.postgresql.org/)

Geonode (http://geonode.org/)

GeoServer (http://geoserver.org/)

OGC services – OpenGeoSpatial (https://www.opengeospatial.org/standards/owc)

React JS (https://reactjs.org/)

Mapstore (https://mapstore.geo-solutions.it/)

Leaflet maps (https://leafletjs.com/)

xamples of references, presented according to the rules of the *Interinstitutional style guide* (http://publications.europa.eu/code/en/en-250904.htm):

List of abbreviations and definitions

RDH	Risk Data Hub
DRMKC	Disaster Risk Management Knowledge Centre
EAV	Entity Attribute Value
ETL	Extract Transform Load
GDACS	Global Disaster Alert and Coordination System
UI	User Interface
DB	Database
OGC	Open Geospatial Consortium
WMS	Web Map Service
WFS	Web Feature Service
JS	JavaScript
JSON	JavaScript Object Notation
ORM	Object Relational Mapping
CSW	Catalogue Service for the Web
API	Application Programming Interface
SQL	Structured Query Language

List of figures

List of tables

KJ-NA-30065-EN-N

The European Commission's science and knowledge service

Joint Research Centre

JRC Mission

As the science and knowledge service of the European Commission, the Joint Research Centre's mission is to support EU policies with independent evidence throughout the whole policy cycle.

EU Science Hub
ec.europa.eu/jrc

@EU_ScienceHub

EU Science Hub - Joint Research Centre

EU Science, Research and Innovation

EU Science Hub

Publications Office of the European Union

doi:10.2760/798003

ISBN 978-92-76-15386-3