

GFDRR - HEV-E

Final Report



Giovanni Allegri
GeoSolutions SAS
02/05/2018
Version 01.01

Table of Contents

[Introduction](#)

[HEV-E mockups](#)

[Interviews](#)

[Data acquisition](#)

[Backend services](#)

[HEV-E backend and frontend](#)

[Conclusions](#)

Introduction

This document contains the description of the final deliverables that were produced for the HEV-E platform and deployed on the its official GitHub repository (<https://github.com/GFDRR/hev-e>).

The following activities were conducted and completed:

- Platform design with mockups for the User Interface and the User Experience to be shared and discussed with the client
- Interviews with people directly or indirectly involved into the project and with stakeholders
- Data schema and data samples acquisition from the three Challenge Fund groups (hazards, exposures and vulnerabilities)
- Development and staging/testing server setup, with backend services (GeoNode)
- Platform's server and client components implementation

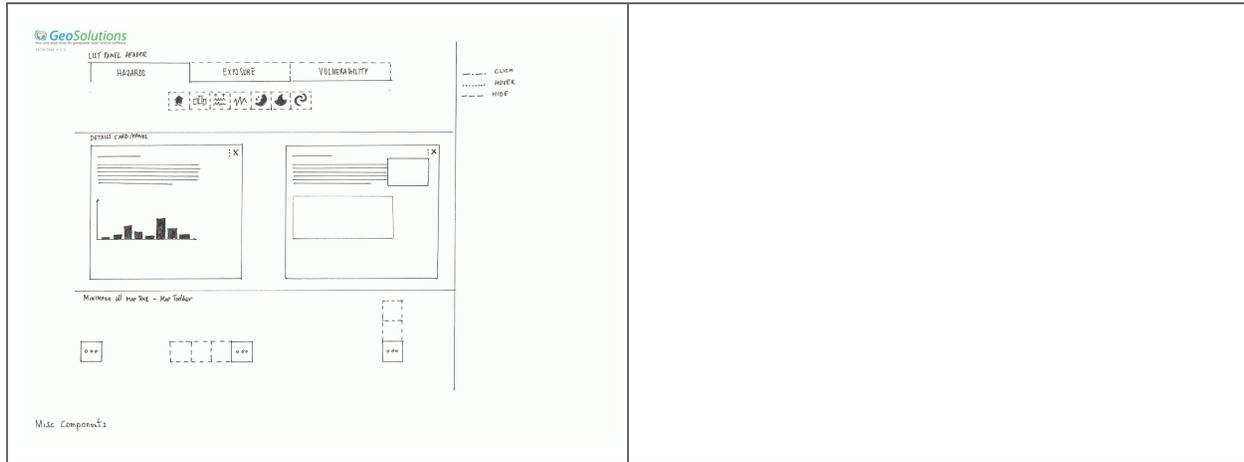


Fig. 1 - HEV-E Mockups

Interviews

A series of interviews has been conducted within January and February 2018 to collect use cases and hints that would have helped the design and the development of the platform. A dozen of people - stakeholders and people involved at various levels in the project - were contacted and interviewed by Webex or Skype calls.

Their answers and feedbacks have been collected and analyzed further during the project team weekly meetings. The relevant outputs have summarised and included within the project's Inception Report document.

NAME	ORGANIZATION	ROLE
Vivien Deparday	WB	GFDRR
Stu Fraser	WB	GFDRR
Mathijs van Ledden	WB	GFDRR
Emma Phillips	WB	GFDRR
Amal Ali	WB	GFDRR
Michel Matera	WB	TTL (task team leader)
Yohannes Kesete	WB	TTL (task team leader)
Sue Loughlin	BGS	Challenge 1 - hazard
Vitor Silva	GEM	Challenge 2 - Exposure

Tiziana Rossetto	UCL	Challenge 3 - Vulnerability
Enrica Verrucci	UCL	Challenge 3 - Vulnerability
Carmine Galasso	UCL	Challenge 3 - Vulnerability

Tab. 1 - Interviewees

Data acquisition

Since February GeoSolutions have been in contact with the Challenge Fund groups to coordinate the acquisition of the hazards, exposures and vulnerabilities data schemas and samples, required for the design and the development of the platform functionalities.

Exposures data have been received from GEM the 16th of February.

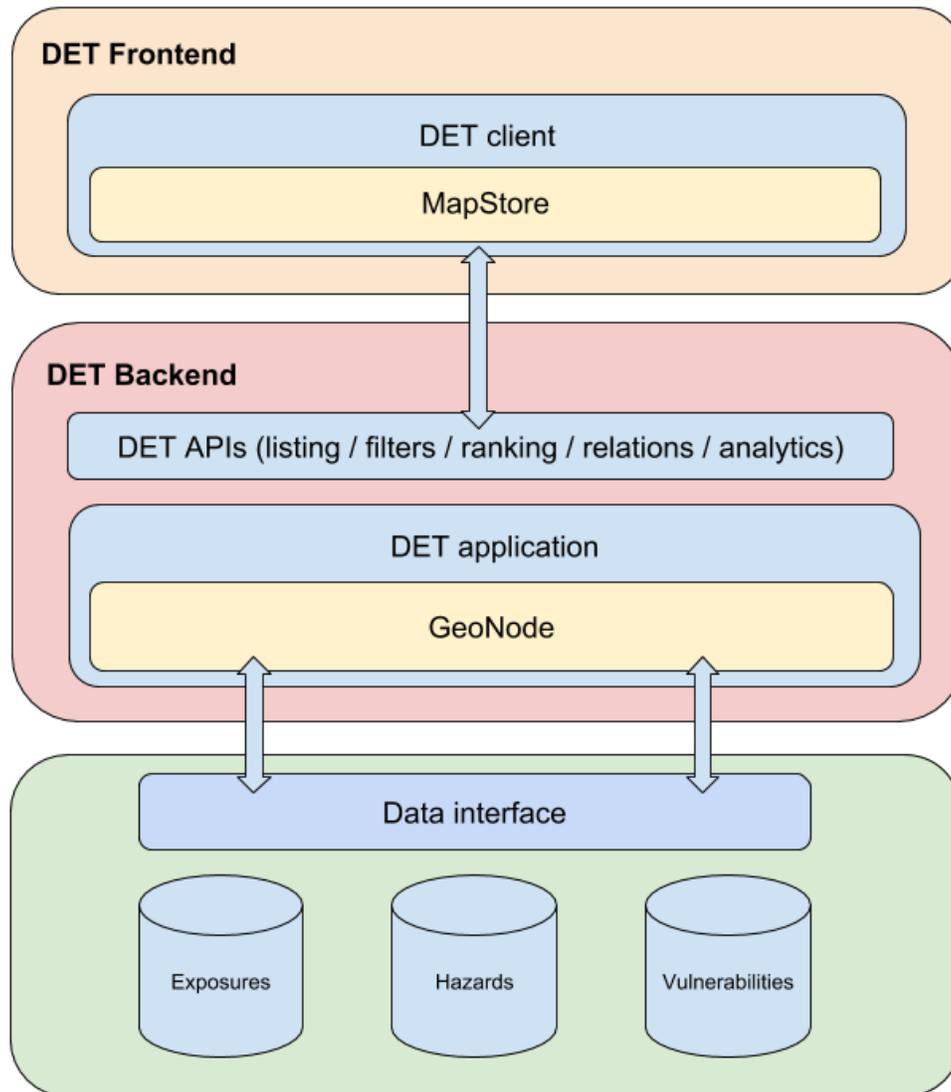
Vulnerabilities Level 3/4 data for vulnerability, fragility and damage to loss functions have been received from UCL the 13th of March.

Hazards data have been received from GEM the 6th of April.

Schemas and available sample data have been preprocessed to make them compatible with the HEV-E backend (RDBMS versions, dump schemas cleanups, etc.) and ingested into the HEV-E Platform using dedicated routines, specifically developed for the purposes of HEV-E. The ingestion code is part of the Platform code that have been officially released.

Backend services

HEV-E is based on GeoNode, according to the following architecture.



A dedicated GeoNode stack of services (PostgreSQL/PostGIS RDBMS and Geoserver) have been setup and configured both for the development and the testing servers. They provide the data tier and the cartographic / webgis services to the HEV-E custom components.

The ingestion phase preprocess and load the hazards, exposures and vulnerabilities data into the backend services, to make them available to the HEV-E application and, through its APIs, to the HEV-E web client.

The original datasets are not organized in the form of simple geographical layers. A lot of work have been put into the preprocessing and ingestion phase to organize and categorize the data as a set of discrete geographical layers, and obtain a consistent approach between the three types of datasets.

HEV-E backend and frontend

A set of server and client components have been developed to provide the specific functionalities of the HEV-E platform.

On the server side a dedicated Django project has been implemented leveraging GeoNode modules and APIs. The Django project implements the backend services needed to ingest data, manage data retrieval from the backend services, implement the HEV-E HTTP APIs for the web client.

On the client side a custom frontend have been developed on top of MapStore, a Javascript / React framework by GeoSolutions dedicated to the creation of advanced web/webgis frontends. The client provides the UI to the HEV-E platform functionalities and guarantees an expandable base for further developments.

The HEV-E components implement the project requirements and in particular:

- Explore the list of available hazards, vulnerabilities and exposures within a selected geographical area.
- Offer a compact preview of the data contents.
- Filter contents by the means of their categories and main distinguishing attributes and by geographical areas (spatial filter).
- Preview geographical contents on the contextual map
- Inspect contents details to obtain the most relevant attributes and informations for the data.
- Download subsets of the data according to the attributes and spatial filters defined by the user. The data is made available both in a simplified form (ESRI Shapefile) and as a Geopackage dump of the original schema and contents.

Conclusions

The staging server is up to date and made publicly available for the review and testing of the platform: <http://hev-e.geo-solutions.it>

HEV-E code has committed to its official Github repository: <https://github.com/GFDRR/hev-e/>.

The same repository will be used to collect feedbacks and issues.

The code will receive any update that will be made to the HEV-E platform for improvements and fixes.