COrruption Risk Indicators in Emergency

Co-funded by the European Union

CORRUPTION RISK INDICATORS IN EMERGENCY

Concept note for the data viz

Deliverable WP 4.4

Alessio Cimarelli - Dataninja srls
Niccolò Salvini - Università di Perugia
Davide Del Monte - info.nodes

Grant Agreement number: 101038790 — CO.R.E — ISFP-2020-AG-CORRUPT

Index of contents

Introduction

The document "CO.R.E. Platform: concept of the back- and front-end" intends to outline the conceptual development idea of the CO.R.E. project platform, as regards the general characteristics of both the back-end, limited to the project team, and the front-end, i.e. the portal on which users can navigate and filter the collected data.

The "CO.R.E. Platform: concept of the back- and front-end" is preparatory to the next activities:

a) Development of the dashboard

Info.nodes will start the development activity as soon as the conceptual note has been approved by the project coordinator and the project partners.

The visualization of the data will follow the guidelines of the team working on the website, in order to obtain perfect integration with the other IT outputs of the project.

The dashboard will be linked with the data lake in order to have an automated data visualization process capable of immediate changes if/when new data is added to the database.

b) Implementation of the dashboard

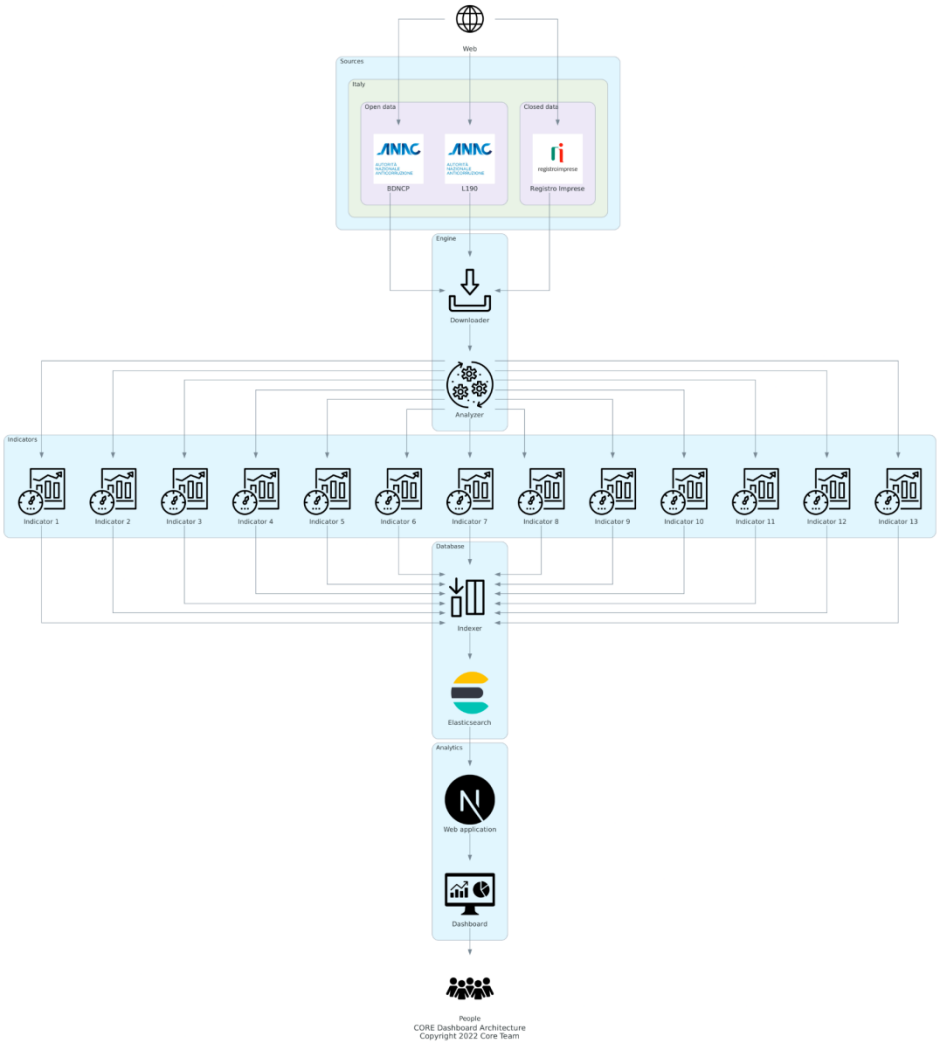
The dashboard will be implemented and integrated into the project website.

Website users will be able to navigate through different infographics and maps showing the main indicators of corruption identified in WP2.

The datasets behind the charts and maps will be made available for download in an open and machine-readable format, in order to enable their re-use for other non-commercial purposes.

Architecture overview

The image below is a rough sketch of the entire system architecture. We will be going over it in a more detailed way by analyzing each element. The purpose of this image is to give a basic idea of the different elements we need to take into account when moving towards the cloud. Different types of data need to be acquired, stored and processed, as is evident (the top web icon). Also, we have different applications that need to interact with each other in order to provide a coherent service to the user. In the end the user interacts with the entire system through a web interface i.e. dashboard, this is the only point of interaction with the outside world.



Data harvesting and analysis

A Big Data pipeline (from now on, data pipeline) is a set of processes that ingest, clean, transform, and store data. Data pipelines are essential for data-driven applications, as they provide a way to reliably and efficiently move data from one place to another. Data pipelines can be batch or real-time, depending on the needs of the application. Batch data pipelines are typically used for ETL (extract, transform, load) processes, where data is extracted from one or more sources, transformed into a format that is suitable for analysis, and then loaded into a target data store. Data pipelines can be complex, with many different components that must work together in order for the data to flow smoothly from one stage to the next. In order to build a data pipeline, it is of the utmost importance to have a clear understanding of the data flow and the dependencies between the various components. In this regard, we discuss the data pipeline for a web-based application that ingests data from a public open data portal + proprietary sources and ultimately serves data to a further web platform i.e. our dashboard. The data pipeline consists of the following components: a web scraper, a data parser, data transformers (2-3 steps), and a data store.

The web scraper (i.e. first part) is responsible for extracting data from the open data portal. For the way dataset are stored there is the need to iterate on hundreds datasets in order to extract them.

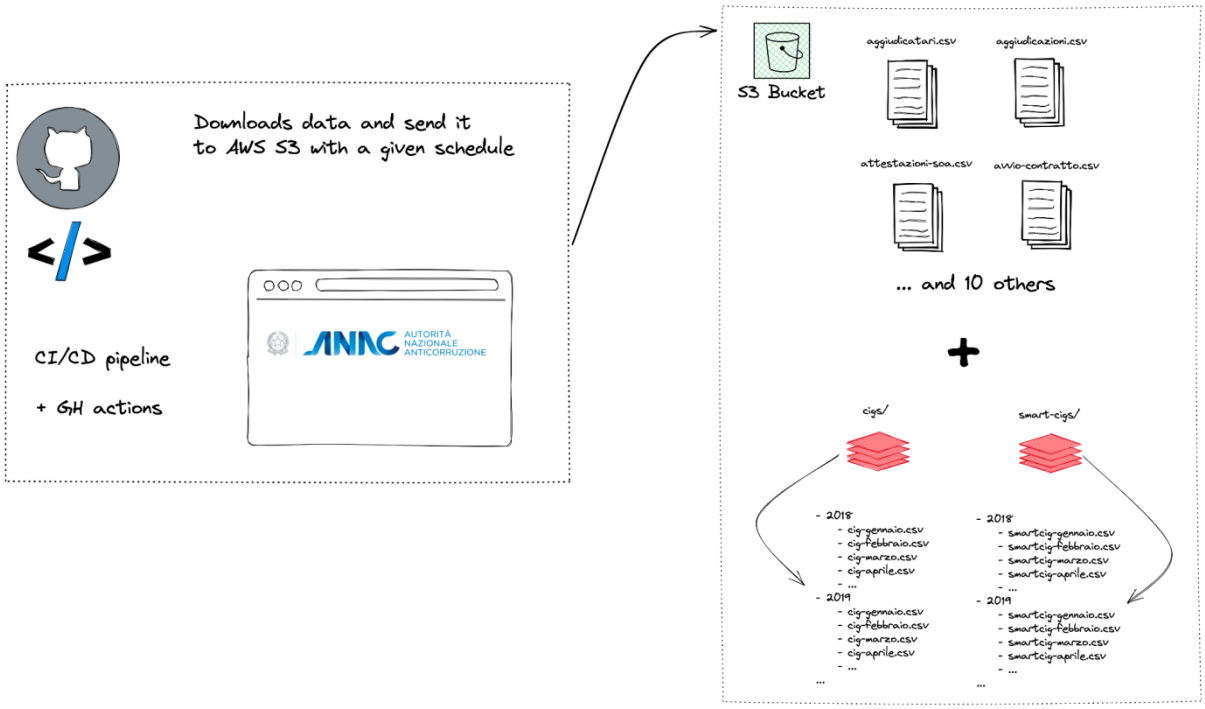
The data parser is responsible for parsing the data into a format that can be consumed by the data transformers. The data transformers are responsible for transforming the data into a format that can be loaded into the target data store. The data store is responsible for storing the data in a format that can be queried by the web application.

Next, the data parser takes the raw data from the web scraper and formats it so that the data transformer can use it. The data transformer then uses a series of transformations on the data, such as filtering, aggregation, and joining. Lastly, the data store is utilized to store the transformed data.

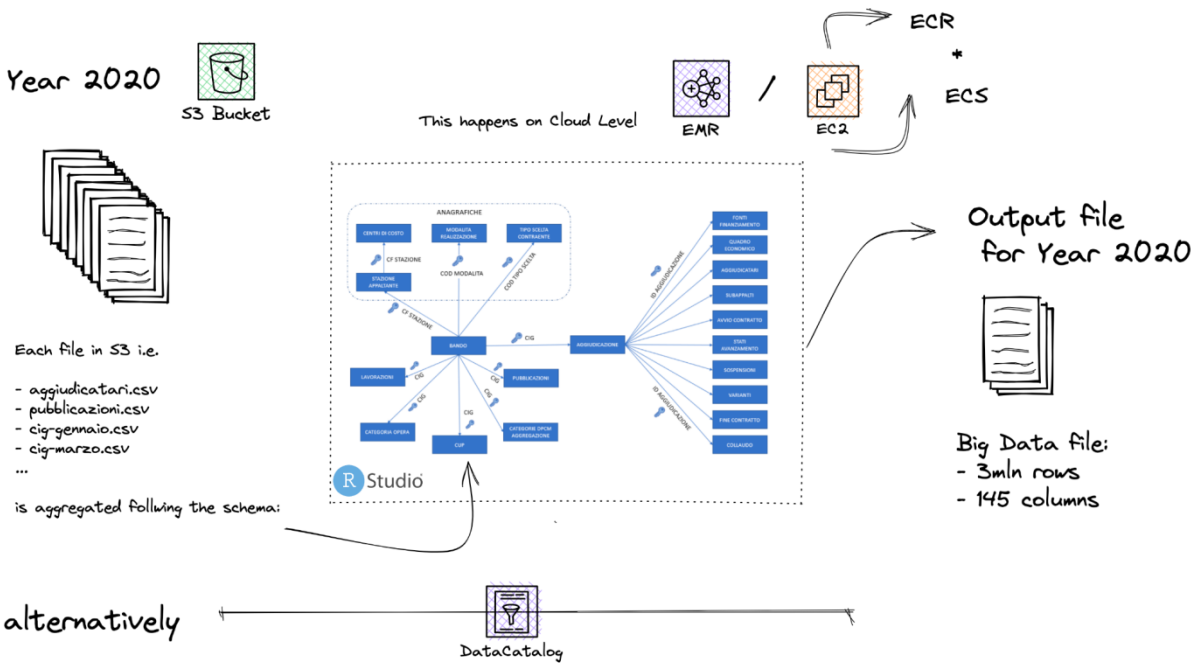
In more details, data needs to be downloaded from dati.anticorruzione. Data as such should be considered as dynamic since every other month at least some part of it is updated, as it appears clear by looking at the metadata information regarding the latest update.

A common choice is to extract a load of data directly into a temporary space, said Data Lake. A data lake is a centralized repository that allows you to store all structured and unstructured data at any scale. You can store data as-is, without having to structure it first, and perform different types of data analysis - from dashboards and visualizations to big data processing, real-time data analysis which ultimately needs some preprocessing

This is also needed since data is updated monthly and any analysis which may follow the month in which we operate the downloading/ingesting phase may prevent the latest data to be considered in our analysis, i.e. indicators calculation.

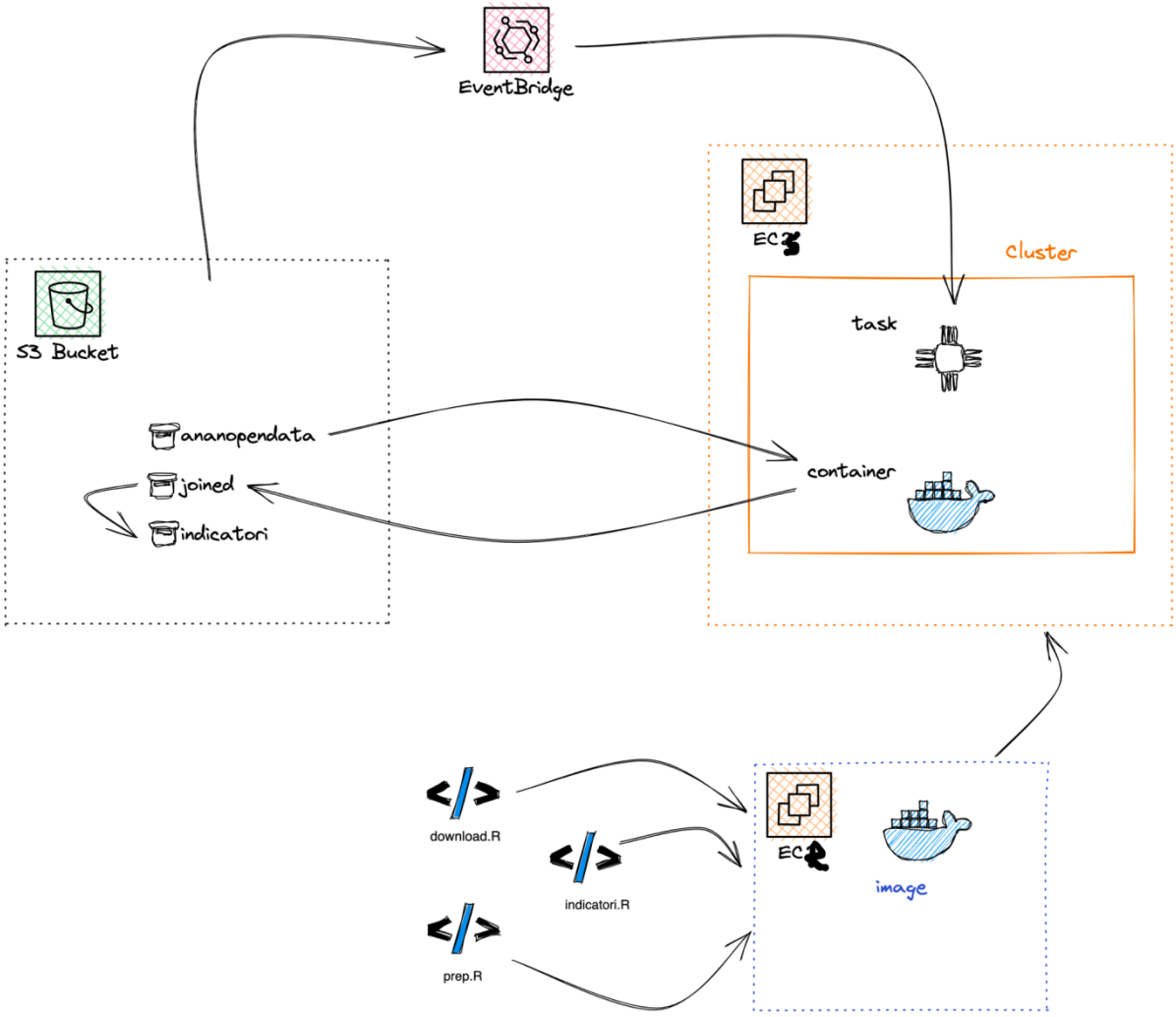


Data up to this point is stored in S3 and needs to be joined according to the database schema provided by the reference. The space requirements for these operations are so large that they cannot be resolved on a local machine. As a result, calculations need to be exported in the cloud, where space and computing power can be adjusted as needed. The image below illustrates the aggregation phase happening for a single year i.e. 2020.



The data coming from yearly data pipelines needs to be concatenated into a larger file, accounting for approximately 10 million rows and 145 columns. Indicators and their composites must then be executed. The frontend requires the indicators to be served as multiple files, partitioned into emergencies and specific target statistical units with respect to the indicator considered.

The AWS cloud services infrastructure is a complex system that can be seen in the figure below. Data is ingested into an S3 bucket on a monthly basis, and then processed through various functions to ensure accuracy and efficiency. The data is then stored securely and can be accessed by authorized personnel. Additionally, the system is designed to be secure and reliable, with regular backups and updates to ensure that the system remains up-to-date.


EventBridge
EC2
Cluster
task
S3 Bucket
ananopendata
joined
indicatori
container
download.R
indicatori.R
prep.R
EC2
image

Indicators database

The analyzer takes raw data in input and generates all possible values of all indicators to be indexed in Elasticsearch following this schema.

Field name	Field type	Sample value
Indicator ID	ID	3
Indicator Name	String	One-shot opportunistic companies
Indicator Value	Number	0.65
Indicator Last Update	Datetime[ISO8601]	2022-11-11T11:11:11Z
Emergency ID	ID	1
Emergency Name	String	COVID-19
Aggregation ID	NAMESPACE:ID	NUTS:ITH3
Aggregation Name	String	Toscana
Aggregation Type	String[enum]	NUTS-2
Country ID	NAMESPACE:ID	NUTS:IT
Country Name	String	Italy
Data Last Update	Date[ISO8601]	2022-10-31

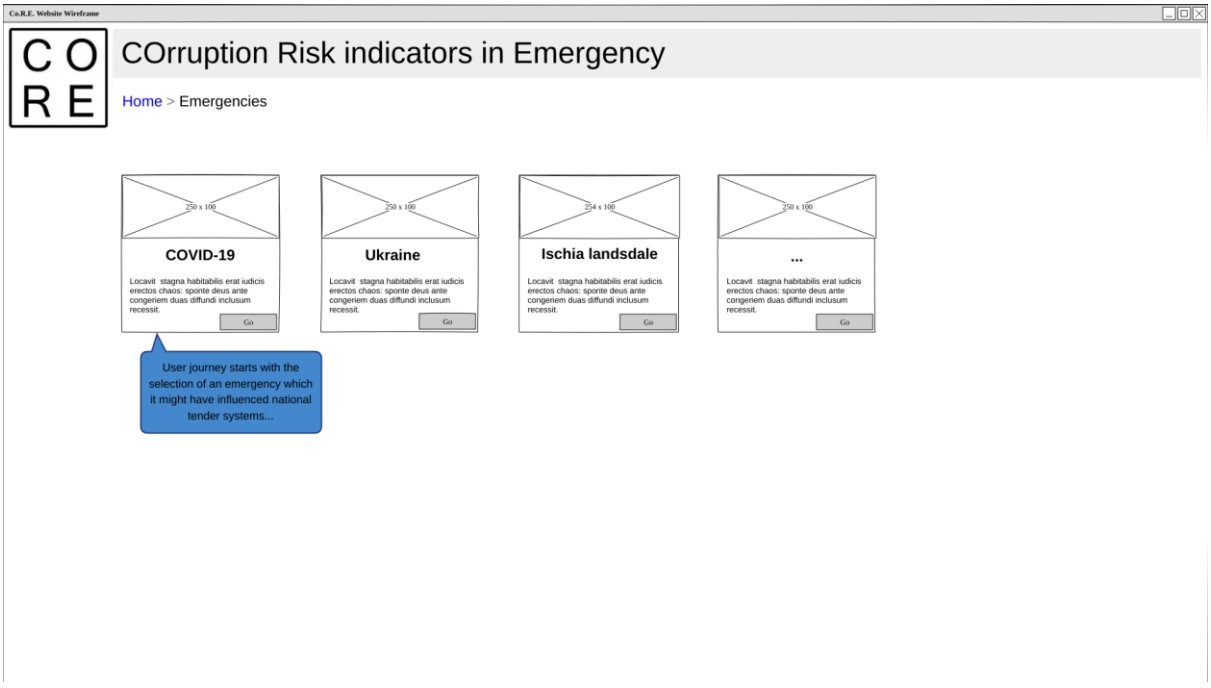
Public dashboard embedded in the project website will use that near-real-time database to offer a rich data exploration experience to users.

Public dashboard

Public dashboard exposes all results of previous analysis and allows users to explore all data dimensions of computed indicators. We propose a user journey from emergencies to corruption risk indicators.

Page 1 - Emergency selection

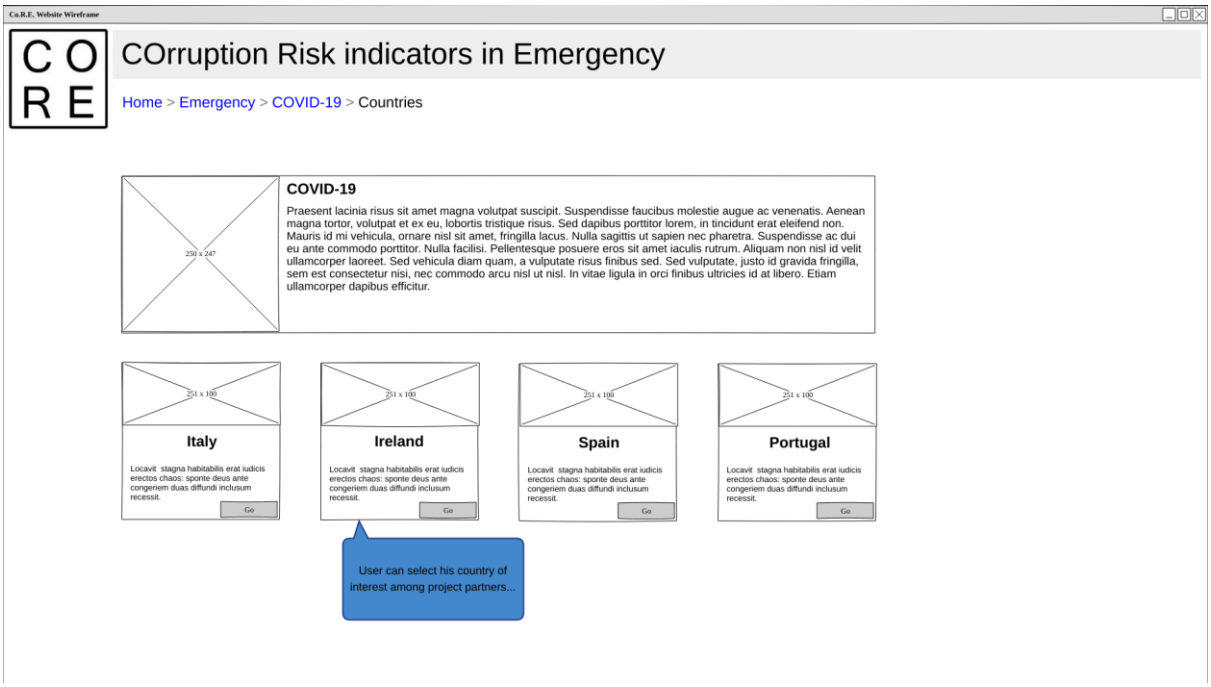
User should first select the emergency he's interested in.



Emergency selection

Page 2 - Country selection

Then the user should select his country of interest.

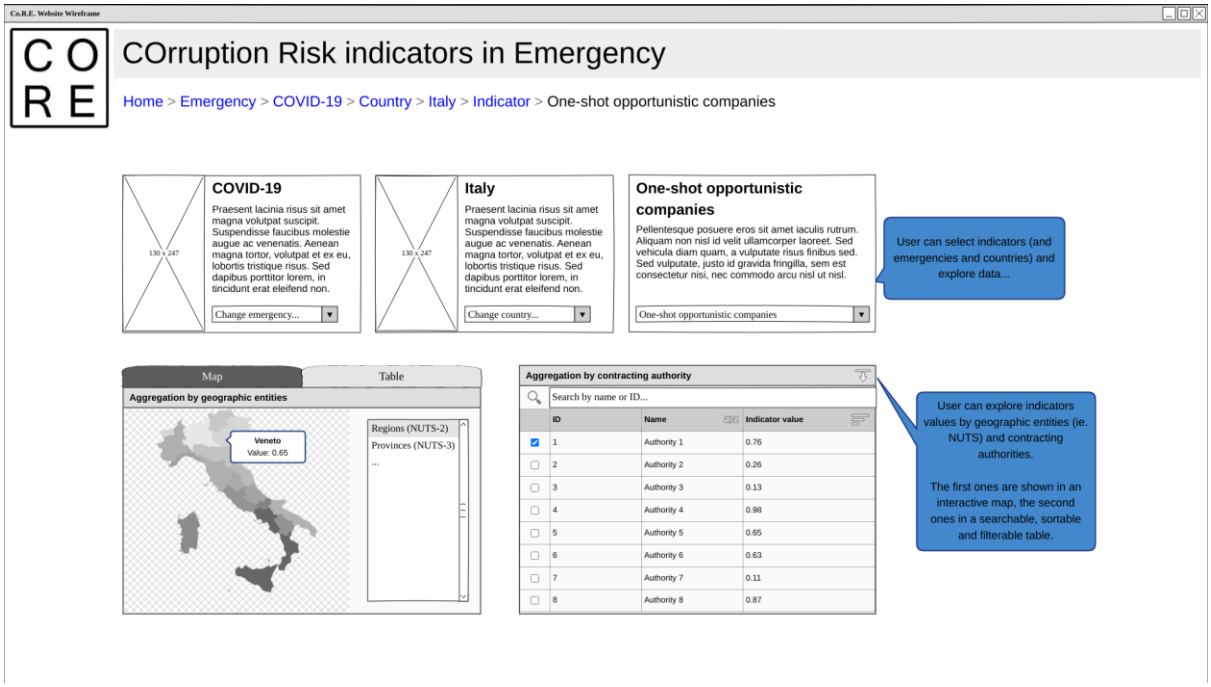


Country selection

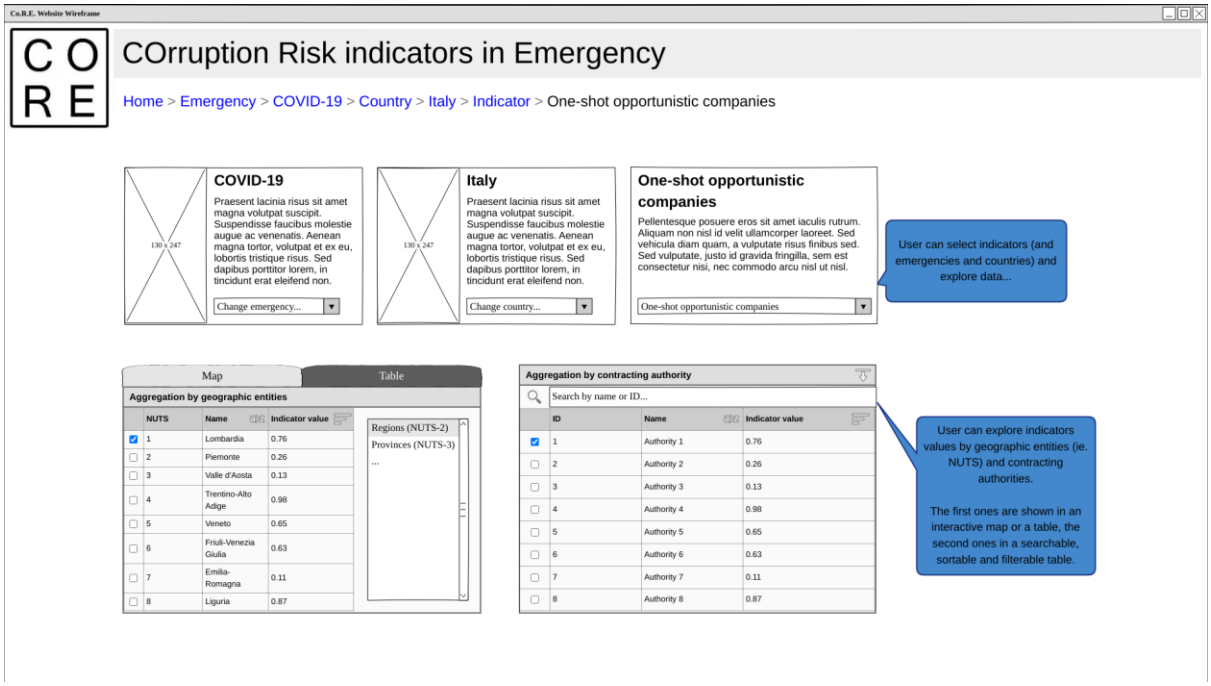
Page 3 - Indicators exploration

Dashboard allows users to explore indicator values by aggregation level: geographic entities (NUTS levels) with map or table and contracting authorities with an interactive table.

All filtered data can be downloaded from map or table widgets and the state of the page (ie. emergency, country, indicator selections, and filters on tables) will be encoded in URL to ease user bookmarking and sharing.



Dashboard with map



Dashboard with table

All software components will be released with open licenses and accessible also from different repositories (e.g. Github).