



Development of a semi-structured database for back-analysis of the foundation stiffness of offshore wind monopiles

Bruno Stuyts^{1,2} · Wout Weijtjens¹ · Christof Devriendt¹

Received: 25 August 2021 / Accepted: 23 March 2022 / Published online: 11 April 2022
© The Author(s) 2022

Abstract

Offshore wind turbines founded on monopiles are highly dynamic structures in which the stiffness of the soil adjacent to the monopile controls the natural frequency of the structure. As the loading regime and ground conditions surrounding the foundation are subject to considerable uncertainty, adaptable digital twins of the offshore structures are valuable as they allow the use of in-field monitoring data for model updating. As soil conditions and water depths are rarely uniform across a wind farm site, each structure is expected to behave differently. To back-analyse structural performance, geotechnical and structural data needs to be retrieved at every foundation location. A serverless cloud-based application was developed to allow quick and reliable storage and retrieval of geotechnical and structural data. The database was combined with an API layer to allow parametric data retrieval for back-analyses and digital twin updating across an entire wind farm. As the web application is hosted in the cloud, the data can be accessed through simple HTTP requests by authenticated users working offshore, in the office or remote. The performance of this solution is illustrated with a case study in which foundation stiffness across an entire wind farm site is parametrically calculated and updated.

Keywords API · Back-analysis · Cloud · Database · Offshore wind

1 Introduction

Offshore wind turbines are widely used in Western Europe to produce renewable energy. Monopile foundations have been the preferred foundation option due to their simplicity and relative ease of fabrication. In 2019, 4258 monopiles had been installed, making up 81% of all installed offshore wind turbine foundations [28]. Because offshore wind turbines are tall and slender structures, a large bending moment needs to be absorbed by lateral soil reactions. In contrast to other geotechnical engineering problems, the

ultimate capacity (ULS) of the foundation is not the governing limit state [22, 23]. Dynamic excitations due to wind and waves lead to accumulation of fatigue damage in the structural steel of the foundation. The natural frequency of the structure governs the period and magnitude of these cycles and the stiffness of the soil next to the monopile is a determining factor for this natural frequency.

Devriendt et al. [10] present monitoring data for offshore wind structures with monopile foundations and the monitored values showed a underestimation of the natural frequency between 5 and 15%. This observation is in line with mismatches reported by Kallehave et al. [16]. To ensure more accurate modelling of the natural frequency, a back-analysis of lateral monopile response across an entire wind farm site is required. This allows new design methods to be calibrated to field measurements and sensitivities of the natural frequency to load level and seabed level changes to be checked. Weijtjens et al. [27] demonstrate that the natural frequency of the foundation can change over time and these changes need to be explained.

Performing back-analysis of pile response across an entire wind farm site requires a large amount of data to be

✉ Bruno Stuyts
bruno.stuyts@vub.be

Wout Weijtjens
wout.weijtjens@vub.be

Christof Devriendt
christof.devriendt@vub.be

¹ OWI-Lab, Vrije Universiteit Brussel, Pleinlaan 2, Elsene, 1050 Brussels, Belgium

² Geotechnics Laboratory, Ghent University, Technologiepark 68, 9052 Zwijnaarde, Belgium

integrated and parametric geotechnical workflows to be created on the basis of this data. This work presents the development of a database and associated cloud-based Application Programming Interface (API) layer for storage and retrieval of geotechnical data and pile dimensions. The advantages of working with a semi-structured database are illustrated for the CPT data which is used as the primary input for soil parameter selection. A case study on the application of the database and API to farm-wide back-analysis of foundation stiffness at the mudline level is presented.

2 Geotechnical database requirements

2.1 Relational databases

Relational Database Management Systems (RDBMS) are widely used for storage of data and modelling of the relations between entities. It forms the basis of the Structured Query Language (SQL) [17] which is a language used for database operation. SQL statements allow data to be retrieved, created, updated or deleted.

Several initiatives have been taken on a national level to develop relational databases for geotechnical data management [18, 24, 26]. These databases are managed by local authorities and have well-defined workflows for collecting new data. The spatial extent of the data is often limited to the jurisdiction of the local authority. Because the write access to these databases is restricted, it was not possible to use them for this work. However, the relations modelled in these databases were taken into account for the development of the database used in this study.

PostgreSQL [25] was used as the RDBMS for this study. It is an open-source object-relational database which supports many data types and can be extended with geospatial functionality as described in Sect. 2.3. It has built-in features for data integrity and security and is widely used across a range of applications.

2.2 Unstructured data

Unstructured databases (NoSQL) which do not model the relationship between entities have become increasingly popular for big data applications due to their increased efficiency [15].

However, while large amounts of geotechnical data may exist when compiling data over several projects, the data volume is still limited compared to social media data [3] or applications where data is continuously monitored [7]. Moreover, the structure of the data is generally well defined, although the format of the data itself may differ. Modelling relations between entities is therefore still

meaningful but efficiency gains can be found by modelling part of the data as unstructured data.

For example, modelling the relations of an individual CPT test to a geographical location, site survey and construction project is meaningful. But each measurement point of the CPT (a test depth value, which usually corresponds to the cone rod length, with corresponding values of cone tip resistance q_c , sleeve friction f_s and pore pressure u_2) does not necessarily need to be modelled as a separate entity. Doing so would greatly increase the number of rows in the corresponding database table. Instead, JSON (JavaScript Object Notation) fields [20] can be used to model the CPT data in an unstructured manner. The measurements for the entire CPT trace are converted into JSON format and written to a corresponding data field in the database (Fig. 1).

The advantage of this approach is that database fields do not need to be known during creation of the database. Several cone types exist which have different outputs (e.g. inclination measurements). Such fields can easily be added to the JSON data without having to update the database structure. For example, when inclination measurements are available, deviations between the true vertical depth and the measured cone rod length due to cone rod inclination can be identified and corrected for. Including inclination data in the JSON can therefore be useful. The disadvantage of JSON fields is that there is a potential difference in the JSON structure which is stored. Preprocessing of the data should be performed to ensure that the same data standards are enforced (e.g. consistent use of z [m] and q_c [MPa] as labels for the depth and cone resistance data, respectively). Equally, data from standardised data transfer formats such as AGS [4] or DIGGS [5] can easily be converted to JSON.

PostgreSQL implements these JSON fields and therefore allows the advantages of unstructured data modelling to be combined with explicit modelling of the important relations between entities.

2.3 Geospatial data

PostGIS [19] is an extension to the PostgreSQL relational database which adds geospatial functionality to the database. Geographic information such as foundation locations (Point fields), project areas (Polygon fields), cable and pipeline trajectories (Linestring fields), ... can be captured. Geospatial queries can be executed (e.g. finding all borehole locations in a 500 meter radius around a central point) which are useful when geotechnical data is not available at the exact location of the foundation.

PostGIS also implements different coordinate systems which makes the database more robust when projects are spread over large geographical distances. The relevant

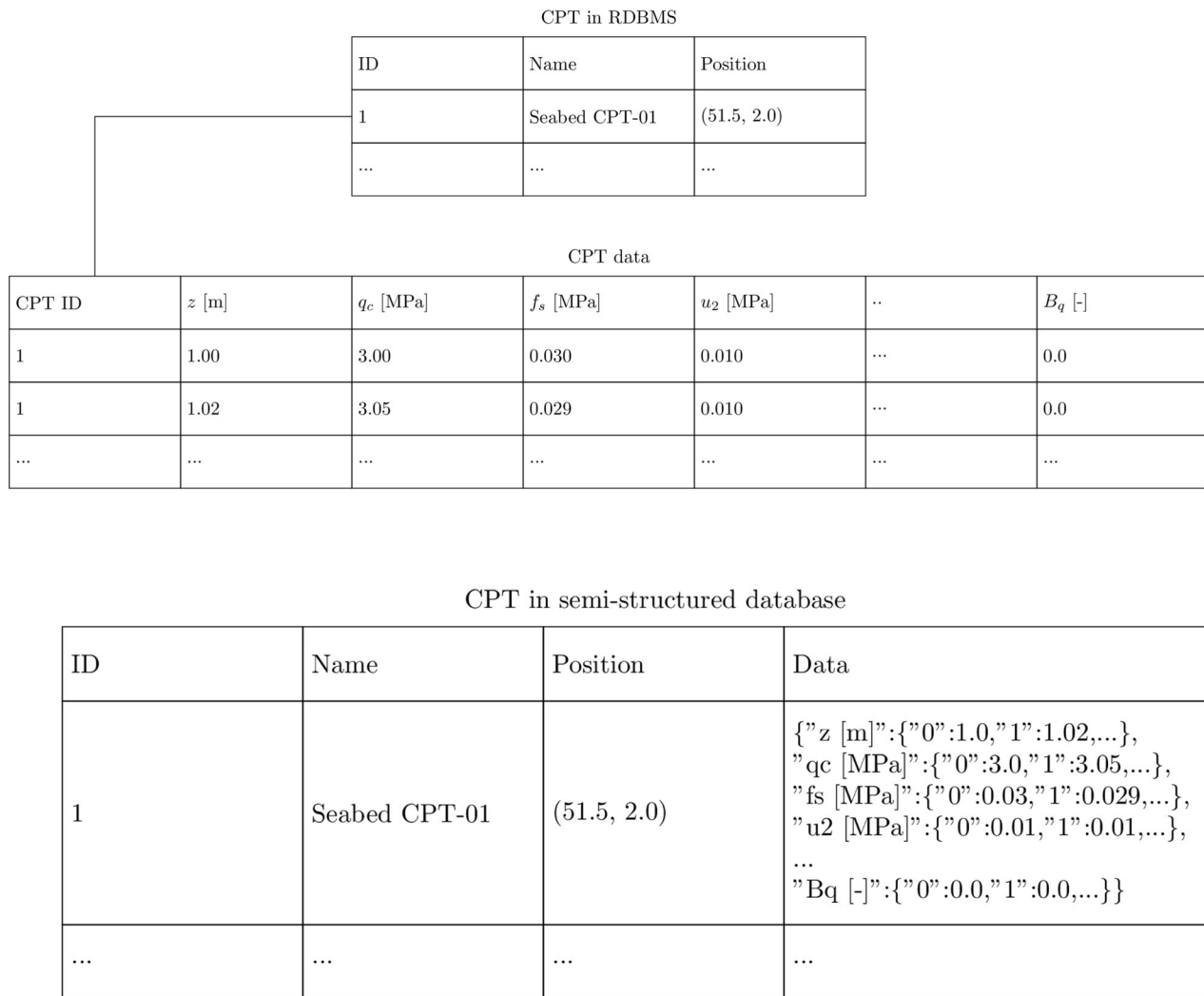


Fig. 1 CPT data storage in RDBSM and semi-structured database

projection can be chosen during data entry and PostGIS performs the coordinate transformations automatically.

Using PostGIS allows for a direct import of the geospatial data stored in the PostgreSQL database into a Geographical Information System (GIS) [29]. Alternatively, an Application Programming Interface (API) can be created to retrieve data from the database in GeoJSON format [8]. The advantages of an API over direct database access are discussed in Sect. 4.

2.4 Interpreted geotechnical data

Raw geotechnical test data is almost always supplemented with an interpretation. In most cases, a layering is established based on the available geotechnical data as a minimum. In addition to the layering, geotechnical parameters are often selected for geotechnical design. These design parameters can be selected based on site-specific data, but

if such data is not available, data from boreholes further away may be taken into account. Capturing which data sources are used for the selection of the layering and geotechnical parameter profiles is a requirement for the database.

Moreover, multiple soil profiles can be defined for a single foundation location. A common use case is a sensitivity study where a lower bound, best estimate and upper bound soil profile are specified.

3 Owimetadatabase: a semi-structured database for geotechnical and structural data

Based on the requirements in Sect. 2, a PostgreSQL database with PostGIS extension was set up. The Django web framework [12] was used to model the relations between

the different entities. A modular approach was followed for the modelling of the database in which three modules were discerned. The data models related to the location of foundations and soil tests are discussed in Sect. 3.1, geotechnical data models are described in Sect. 3.2 and the structural properties of the foundations are included in Sect. 3.3. The modular application architecture allows for new modules with their own entity relations (e.g. fatigue details, secondary structures, ...) to be added. The database is called owimetadatabase, since it contains metadata on all foundations being monitored by OWI-Lab.

3.1 Location data

The data model for the locations is shown in Fig. 2 as an UML (Unified Modelling Language) entity relation diagram. As most construction works are organised on a project basis, the *Project* is the primary entity. A *Project* is described by a title, description and a project area. The project area is defined as a multiple polygons as a project can be composed of several sub-areas. Project-level permissions are organised by defining a many-to-many (M2M) relation between *Projects* and *Permission Groups*. A *Permission Group* is a collection of users having certain permissions and access to a project can be restricted by adding only those groups which are allowed to see the project data. Because it is not possible to foresee all data that might need to be captured when creating the data, a JSON field for additional data is used to capture any other data the user may want to capture. Each data model also has fields to log the date and time of creation and last update and the user that have created and last updated the entries. These fields are not shown in the figure.

Because foundation locations and site investigation locations share many common properties, a base model (*Location*) has been created which describes the general aspects of a location. A *Location* is uniquely linked to a *Project* using a foreign key (FK) relation. The diagram

shows that a *Project* can have several *Location* instances, whereas each *Location* only belongs to one *Project* instance. A *Location* is defined by a title, a description and its geographical position (a *Point* field describing the latitude and longitude). The elevation is also included. For offshore foundations, this is the water depth relative to Lowest Astronomical Tide (LAT).

Foundation locations are modelled in the *AssetLocation* table. *AssetLocation* has a foreign key to *Location*. Each *AssetLocation* corresponds to a single *Location* but a *Location* can either be an *AssetLocation* or a geotechnical *TestLocation* (Sect. 3.2).

3.2 Geotechnical data

The entity relation diagram for geotechnical site investigation data is shown in Fig. 3. Geotechnical data is gathered during a *Survey Campaign* on a given *Project*. Using a foreign key relation, multiple survey campaigns can be performed on a project. Each survey has a start and end date. Surveys can consist of several *Test Locations*. Each *Test Location* instance is related to one *Location* instance. At a *Test Location*, multiple sources of geotechnical data can exist.

In-situ Test instances can be CPTs, SPTs, pressuremeter tests, ... These in-situ tests are characterised by their *In-situ Test Type*. The start and end depth for the in-situ test are recorded as well as the date of the test. The in-situ test conditions, raw test data and processed test data are recorded in JSON fields. As mentioned in Sect. 2.2, these JSON fields offer flexibility in terms of the format in which the test data is provided and stored. Using this architecture, data from several contractors and equipment types can be handled in the same general framework.

At a *Test Location*, *Geotechnical sample* instances can exist which are defined by the top and bottom depth of the sample. A *Sample Type* needs to be selected (e.g. push sample, hammer sample, ...). Certain samples also have

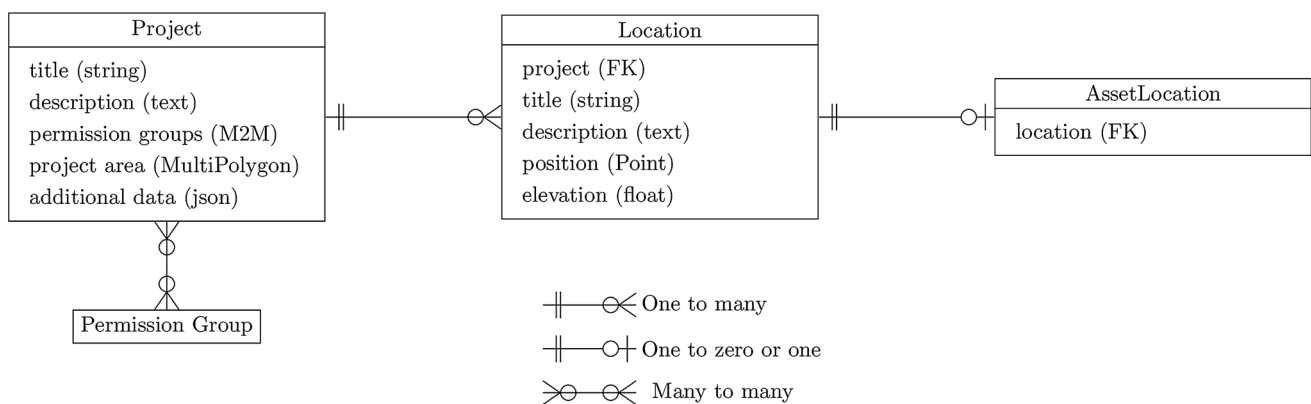


Fig. 2 Data modelling for locations

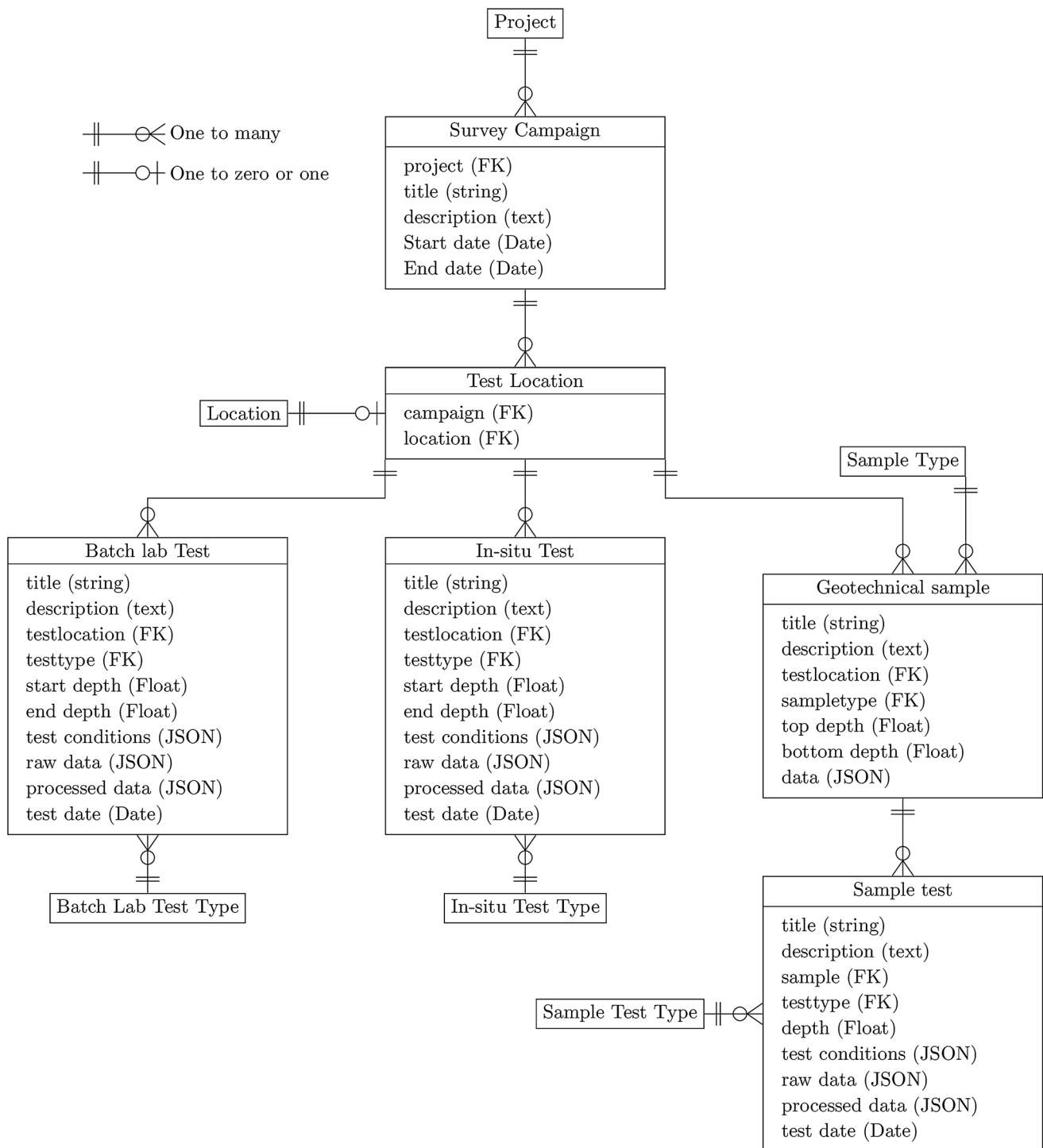


Fig. 3 Data modelling for geotechnical site investigation data

associated data (e.g. SPT samples with the number of SPT blows) which can be stored in a JSON field. *Sample tests* (laboratory tests) can be performed on a *Geotechnical sample*. For a *Sample test*, the depth of the tests as well as the *Sample Test Type* is recorded. Test conditions, raw test

data and processed test data are again stored as JSON fields.

In some cases, especially when processing data from historical surveys, modelling each sample separately may not be desirable for simple lab tests. *Batch lab tests* are lab tests which are performed in bulk on samples from a single

borehole. For example, water content tests are often performed on-site and are reported per borehole. The *Batch lab Test* data model allows such data to be captured efficiently with data from multiple tests being combined in the test conditions, raw data and processed data fields.

The entity relation diagram for soil profiles containing interpretations of geotechnical data is shown in Fig. 4. A *Soil Profile* instance is related to a *Location* through a foreign key relation. Note that a *Soil Profile* instance can both be connected to an *Asset Location* and a geotechnical *Test location*. This reflects the reality where geotechnical investigation data is not always available at the exact foundation location. Each *Location* can have multiple *Soil*

Profile instances, which caters for the case where multiple design scenarios (e.g. lower bound, best estimate and upper bound) need to be checked.

Each *Soil Profile* instance is based on a selection of *In-situ Tests*, *Sample Tests* and *Batch Lab Tests*. Many-to-many relations allow the user to specify which tests were used for the derivation of the *Soil Profile*. To allow use of the database for onshore applications, a field for the phreatic water level was included.

Each *Soil Profile* instance is composed of one or more *Soil Layer* instances. These layers have a top and bottom depth and a *Soil Unit* is selected for each layer. For each *Soil Unit*, the dominant *Soil Type* is selected (e.g. sand,

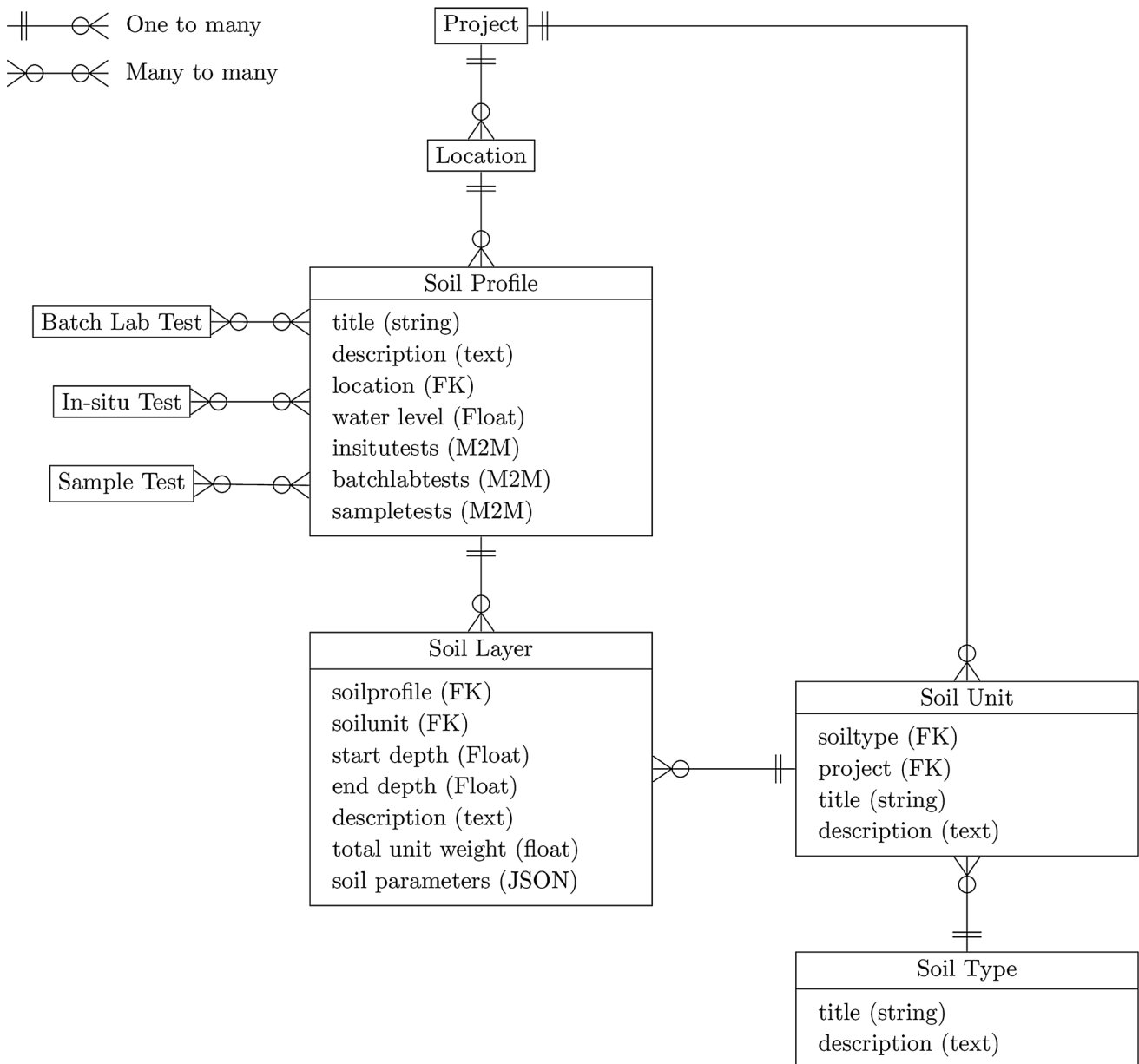


Fig. 4 Data modelling for geotechnical soil profiles

clay, silt, ...). *Soil Units* have a foreign key relation to a *Project* to allow unit-specific data to be retrieved for a *Project*.

Given the importance of the unit weight for vertical stress calculations, the total unit weight can be specified for each layer. Since it is not possible to know a-priori which soil parameters are required for which application, the soil parameters can be added as a JSON field. In this field, the value of the soil parameters in the specified layer is defined. The soil parameters stored in a *Soil Layer* can be used in further design calculations.

3.3 Structural data

The dimensions of the foundations can vary across a project and in some cases, pile diameters and wall thicknesses vary with depth. Moreover, foundation dimensions may change as projects transition from the Front End Engineering Design (FEED) stage to detailed design and finally fabrication stage. The entity relation diagram for the foundations shown in Fig. 5 captures these aspects for the foundation data model. Each *Project* can have multiple *Model Definitions* (FEED, Detailed design, as-built) with corresponding *Foundation* instances. Moreover, an *Asset Location* can have multiple *Foundation* instances (e.g. jacket piles) which have an x- and y-offset from the central position. The vertical position of the foundation and its reference (e.g. LAT) are also defined as fields of the *Foundation* data model. Each *Foundation* is of a given *Foundation Type* (foreign key relation) which can be a monopile, jacket foundation, suction bucket, ...

owimetadatabase is focused on tubular piles but could be extended to other foundation types in the future. A tubular pile is made up of several *Tubular Section* instances. These are effectively the cans of a pile foundation. Each of these *Tubular Sections* has a top and bottom diameter which can be different, so that conical transitions can be modelled. A wall thickness is assigned to each *Tubular Section* as well as a *Material*. The *Material* (e.g. steel) is defined using its density and elastic properties (Young's modulus and Poisson's ratio).

Combining the data models for location data, geotechnical data and structural data allows for farm-wide location-specific foundation response analysis.

4 REST API for owimetadatabase

While the requirements for the database structure are clear, making data available for end users requires the deployment of the database to a (cloud-based) server. Commercial geotechnical database applications mainly make use of project-specific, file-based databases which have limits in

terms of security and availability. Deploying a geotechnical data management system as a cloud-based application can overcome these limitations while offering superior data accessibility. This section describes the cloud-based architecture of owimetadatabase.

4.1 Application architecture

The database solution and associated web application are deployed in the Amazon Cloud (AWS). In contrast with a conventional web application which has a (virtual) server continuously listening for incoming requests, a serverless architecture was adopted [6]. In a serverless application, the code for the web application is hosted on a cloud platform and the owner of the project is not responsible for server maintenance. A cached copy of the code is activated when a request comes in. In AWS, such a serverless application is called a Lambda function. Other cloud platforms offer similar functionality (cloud functions on Google Cloud and Azure functions on Microsoft Azure). The Lambda function does not store any data, so all application-specific data needs to be persistently stored in a database to allow it to be retrieved.

The application architecture is shown in Fig. 6. All cloud functionality is contained in a Virtual Private Cloud (VPC) which ensures that the database and code are securely shielded off from the outside. Access to the Lambda foundation is governed by an API Gateway, which routes the request from the user to the Lambda function. The Lambda function is then allowed to exchange data with the PostgreSQL database (as they are both in the VPC). Note that database access from outside the VPC is impossible, making this a secure solution for data storage.

Since the Lambda function is a cached copy of the code, it can be invoked many times, so sudden bursts of activity do not require larger servers to be deployed. Moreover, in periods with no activity (when users are not interacting with the database), there are no costs associated with the Lambda function. Since owimetadatabase is an application with sudden peaks in traffic and calm periods in between, the serverless application architecture is a cost-effective solution with low maintenance requirements.

4.2 Data security

In Sect. 4.1, the security features of the application itself are highlighted (Lambda function and database running in a VPC). However, additional permissions need to be enforced on the data itself. Different users have different permissions depending on their roles. Writing access to the database is restricted to database administrators who are responsible for the quality of the data included in the database. They ensure that all new records going into the

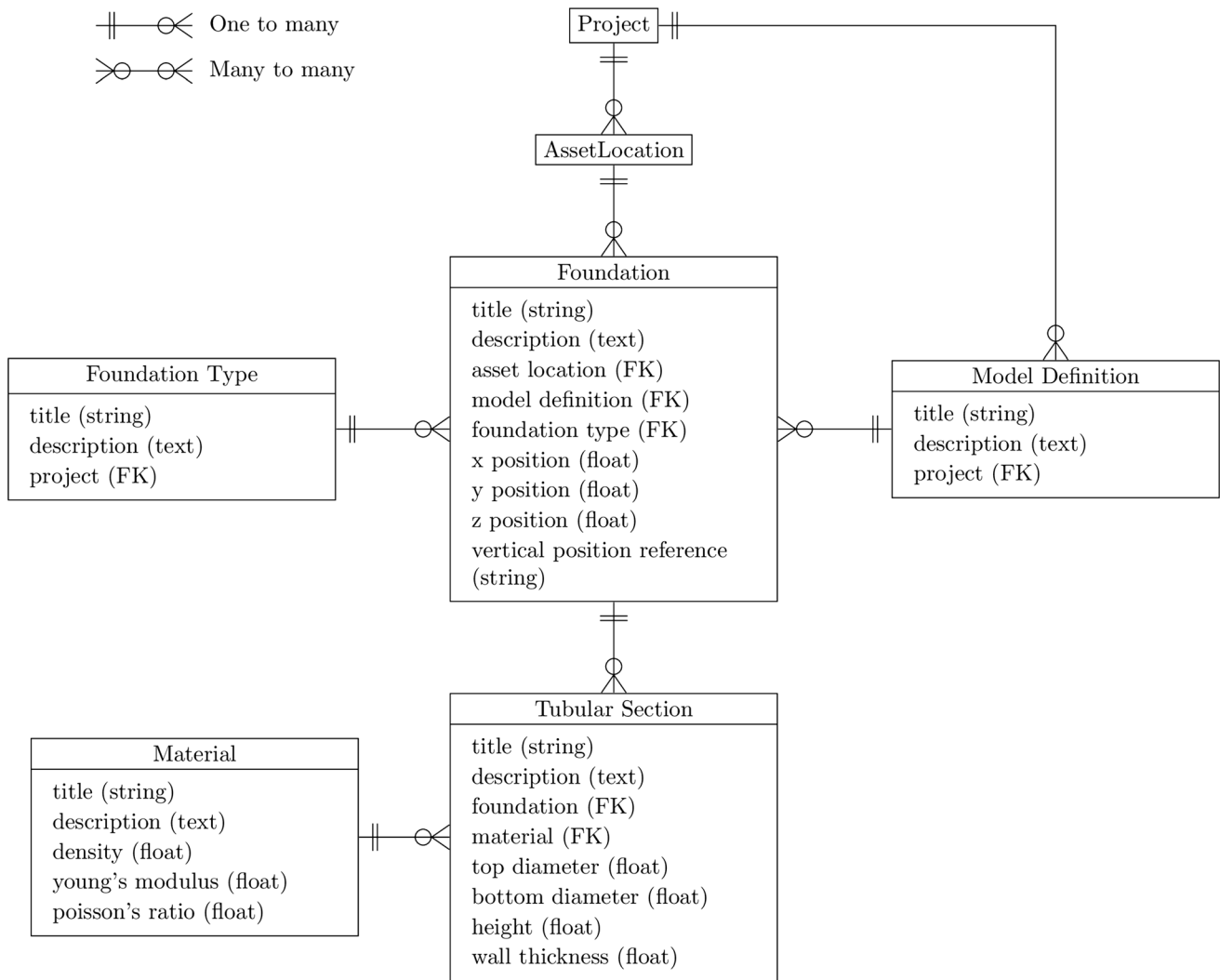


Fig. 5 Data modelling for structural foundation data

database have been quality checked and confirm to the expected standards, especially when JSON fields are involved.

The majority of users only have read access and they are only given access to specific projects. Owimetadatabase hosts proprietary data from offshore wind farm developers which needs to be protected from unallowed access. When a user sends a request, the project-level permissions are checked and if the user has no access to the specified project, no data is returned.

Access rights on the level of data models is also enforced. For example, users who only need factual geotechnical data are not given access to soil profiles and foundation dimensions. Vice-versa, users only interested in the structural properties of the foundations are not given access to the geotechnical data.

4.3 Parametric data using a REST API

REST stands for Representational state transfer (REST) and is a software architectural style that allows data representations to be transferred through an Application Programming Interface (API) [11]. REST APIs allows authenticated users to retrieve data from the database in a standardised format. Figure 6 shows that users are not allowed to connect directly to the database. Direct connections to a database should always be avoided as this carries significant risks for the integrity of the data contained in the database. The web application contained in the Lambda function checks the access rights of the user and defines a REST API which allows users to parametrically retrieve data.

In the web application of owimetadatabase, the API for the data is defined. By sending requests to specific URLs (endpoints), the user can retrieve the desired data. URL

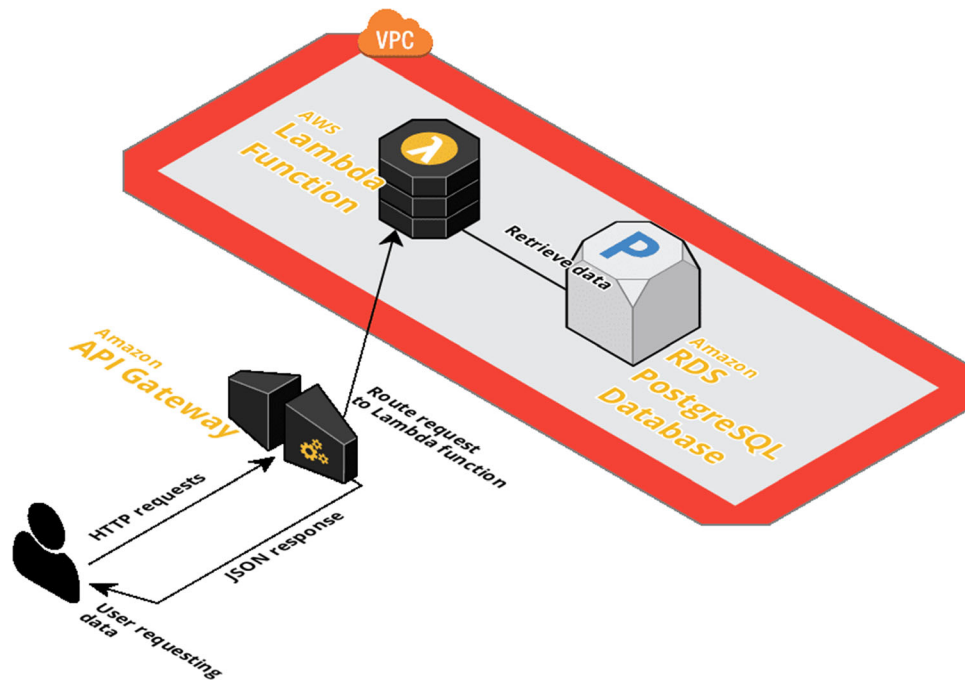


Fig. 6 owimetadatabase application architecture

parameters can be specified to filter the data to only the desired subsets (e.g. only Sample Tests data from Bender Elements tests).

An UML sequence diagram for data retrieval through the API is shown in Fig. 7. The data flow can be illustrated for the case of a user requesting the location of foundations for a specific project (Project X).

The workflow is initiated by a the user sending an HTTP GET request to a specific URL (`/locations/asset-locations?project=Project X`). The API defines the `assetlocations` endpoint allowing users to retrieve a list of `assetlocations`. One URL parameter `project` is defined, containing the title of the project for which data is requested. The request passes through the API Gateway shown in Fig. 6 and reaches the web application contained in the Lambda function. The permissions of the user are first checked. If the user is a registered user and has access to Project X the workflow continues, otherwise, the user gets a message that access was denied.

The web application then creates the appropriate SQL query for retrieving all location data from Project X from the database and sends this query to the database. The database responds with a `queryset` containing the requested data. The data returned from the database is not yet in JSON format. The web application contains the necessary functionality to serialise the data into JSON format before it is returned to the end user. Finally, the user receives the JSON data as an HTTP response and continues using it in a geotechnical and/or structural analysis workflow.

The definition of an API layer for interacting the database allows security to be enforced while providing a standardised and parametric interface for users. Note that a light-weight dashboard website was also developed within the web application to allow users to interactively search and visualise data in the browser.

5 Parametric workflow for foundation stiffness of offshore wind turbine monopiles

The data retrieval described in Sect. 4 was applied to the problem of deriving a stiffness matrix for the foundation response analysis of offshore wind turbine monopiles [2]. The monopile lateral displacement and rotation at the seabed level can be related to the applied shear force and bending moment through a pile stiffness matrix at the mudline (Eq. 1).

$$\begin{bmatrix} F \\ M \end{bmatrix} = \begin{bmatrix} K_L & K_{LR} \\ K_{LR} & K_R \end{bmatrix} \cdot \begin{bmatrix} u \\ \theta \end{bmatrix} \quad (1)$$

The pile stiffness matrix at the mudline can be derived from 1D nonlinear beam-column response analysis. Igou et al. [13] describe the required 1D response analyses to calculate the components of the stiffness matrix consisting of the following analyses:

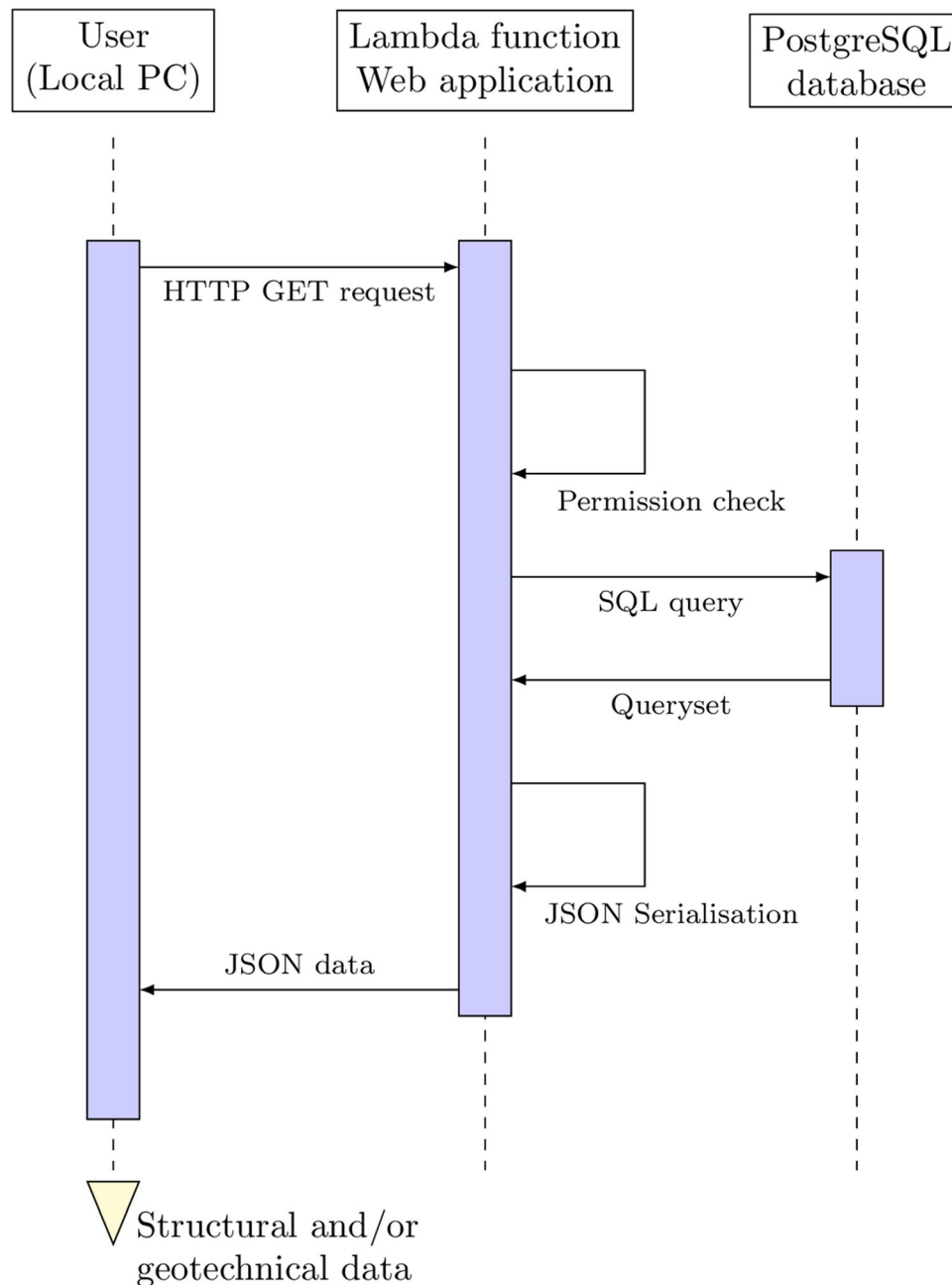


Fig. 7 Sequence diagram for data retrieval from owimetadatabase

- A response analysis with applied lateral force F and restricted pile head rotation θ to derive the lateral stiffness coefficient ($K_L = F/u$);
- A response analysis for calculating to moment M that leads to zero rotation θ for a given pile head displacement u . This defines the off-diagonal term ($K_{LR} = -M_{\theta=0}/u$);
- A response analysis with zero pile head deflection u and an applied bending moment M . The rotational stiffness coefficient can be derived from the observed pile head rotation ($K_R = M/\theta$).

Since the 1D response analysis is nonlinear, the magnitude of the stiffness coefficients depends on the applied load level. Typical production load levels were used for the analysis of the stiffness coefficients.

5.1 Geotechnical data

A farm-wide analysis of the stiffness coefficients was undertaken for a North Sea offshore wind farm with water depths ranging from 21m to 38m. A cross-profile through the windfarm site is shown in Fig. 8. This cross-profile was

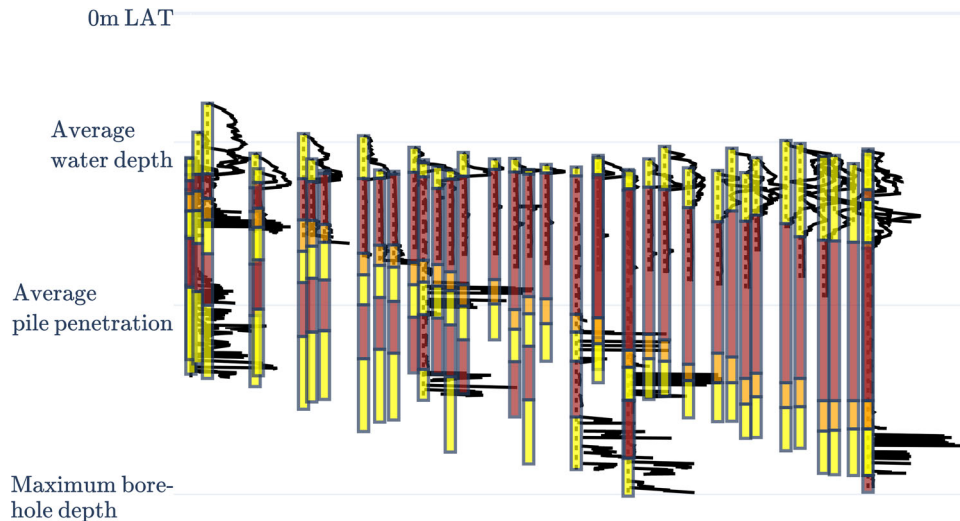


Fig. 8 Cross-profile through the offshore wind farm showing sand (yellow), clay (brown) and sand/clay (orange). Cone tip resistance is overlaid on the soil profiles (black lines)

generated using API calls to the `/insitutestprofile/` and `/soilprofileprofile` endpoints. To define the profile line, the coordinates of the two points defining the profile line need to be defined. The width of a search band around the profile line is also defined as a parameter to also consider test locations which are not perfectly located on the profile line. The API call for in-situ test retrieval is then made using user-specified values for `<latitude1>`, `<longitude1>`, `<latitude2>`, `<longitude2>` and `<offset>`.

```
/insitutestprofile?lat1=<latitude1>&lon1=<longitude1>&
lat2=<latitude2>&lon2=<longitude2>&offset=<offset>
```

The API call returns a listing of the in-situ tests along a profile. To limit the size of the data transferred, only the metadata is returned. To obtain detailed tests results (full CPT data in this case), a call to the `/insitutestdetail/` endpoint needs to be made for each of the in-situ tests returned. Since we know the primary key (pk) of each of the tests along the cross-profile, this primary key can be specified as the URL parameter. We can loop over all test locations along the cross-profile to retrieve the detailed CPT data as JSON.

```
for insitutest in insitutests_along_profile:
  /insitutestdetail?pk=<primary key of insitutest>
```

A similar workflow is followed for soil profiles and all data is processed in Python to obtain the chart shown in Fig. 8.

The soil profiles consist of holocene and pleistocene sand of varying thickness (between 1 and 24m) overlying a thick layer of stiff overconsolidated clay. The stiff clay layer dips towards the right-hand side of the profile and increases in thickness. Below the stiff clay layer, an interbedded sand/clay layer of limited thickness is found overlying a layer of very dense sand with cone resistance $>70\text{MPa}$.

In Fig. 8, cone tip resistance is scaled and is overlaid on the cross-profile to show the locations and depths with highest cone resistance. The low cone resistance in the clay layers and higher cone resistance in the sand layers is clear from the figure.

64 CPT tests were available in the owimetadatabase for the offshore wind farm and these were combined with available laboratory testing and CPT-based correlations to establish best estimate soil parameter profiles at each foundation location. These best estimate soil profiles were uploaded to the database for further use in the workflow. The required soil parameters for sand and clay for the API [1] and PISA [9] methods were selected. For the API methods, these are the relative density class (assessed from CPT according to [14]) and soil type for sand and the undrained shear strength and the shear strain at half of the maximum deviatoric stress in a UU test (ϵ_{50}) for clay. For PISA, the small-strain shear modulus G_{max} was the most important parameter to assess. This parameter was assessed based on in-situ S-PCPT tests, bender element tests and correlations with CPT results [21]. All test results were available in owimetadatabase.

5.2 Structural data

Each monopile geometry for the offshore wind farm was encoded in the database. All monopiles have an outer diameter of 5m with varying wall thickness. The greatest wall thicknesses are observed in the section where the highest bending moments are expected. Pile penetrations range from 28m to 36m. 19 monopile geometries are shown in Fig. 9 with their positioning relative to the water level and the mudline level. The individual sections are visible. A conical section is noted approximately halfway along the water depth. The deepest pile penetrations appear to be correlated with the deepest water depths.

5.3 Pile stiffness matrix at the mudline

The workflow for the derivation of the pile stiffness matrix at the mudline is shown in Fig. 10. For each foundation location, a best estimate soil profile was first created by retrieving CPT, laboratory test data and S-PCPT data from the database. After checking, the interpreted soil profiles were uploaded back to the database. Next, these profile with soil parameters and the pile dimensions were retrieved from owimetadatabase using HTTP requests to the appropriate API endpoint.

Soil and pile parameters were interpolated to a 1D grid and used to calculate nonlinear response curves according to the API and PISA methods. Finally, the stiffness matrix coefficients were derived from 1D nonlinear response analysis with shear forces and bending moments for production loading applied at the mudline level.

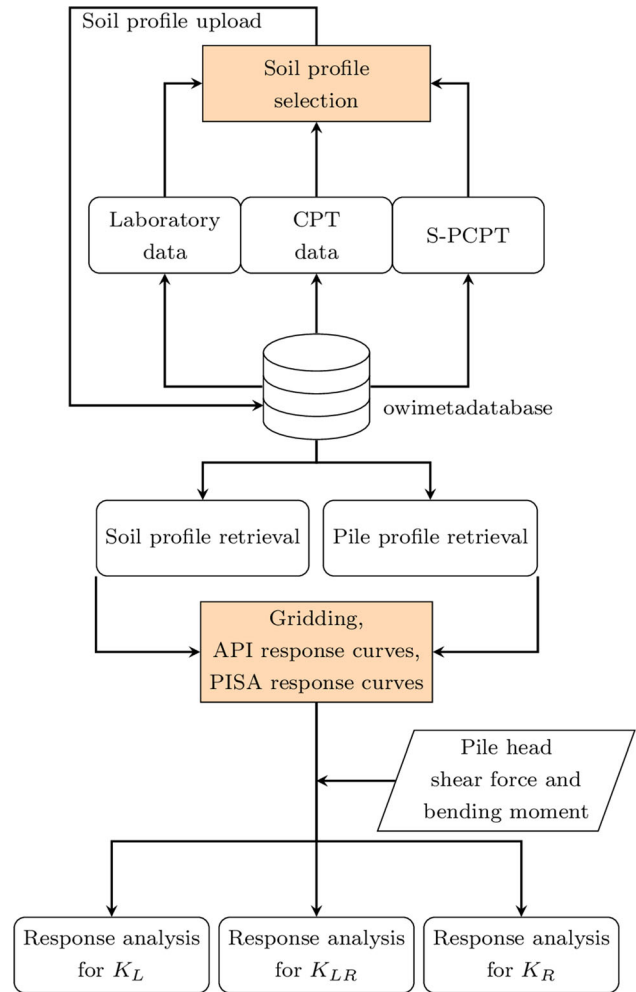


Fig. 10 Workflow for the derivation of pile stiffness matrix at the mudline

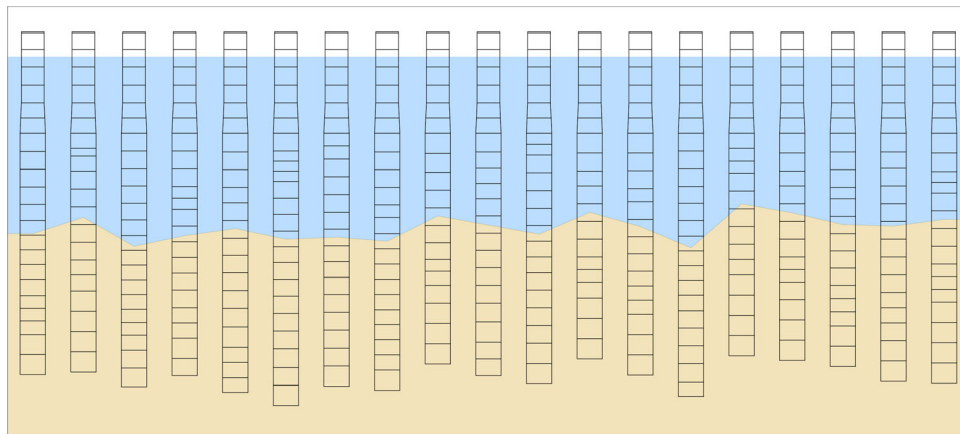


Fig. 9 Example monopile geometries with their water depth and embedment

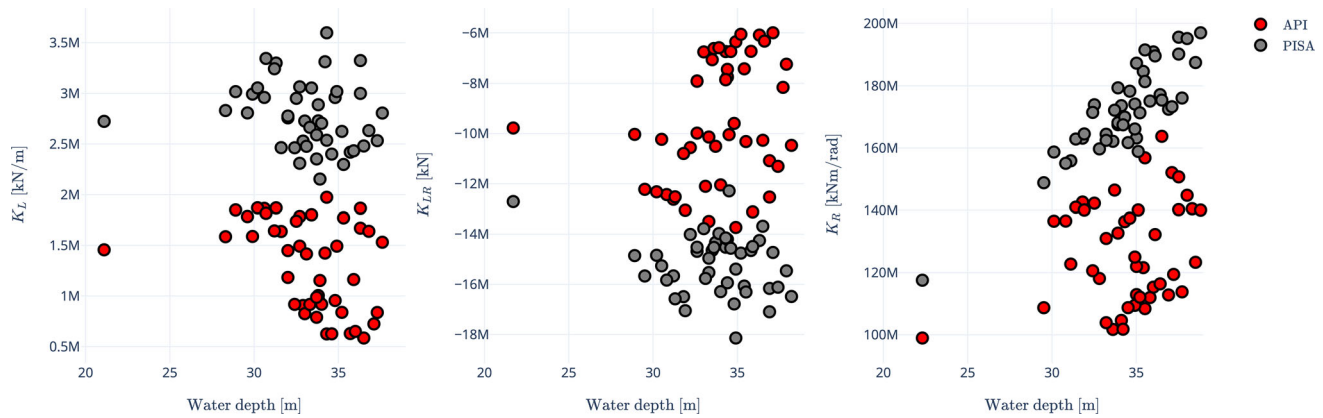


Fig. 11 Stiffness coefficients vs water depth for the entire wind farm

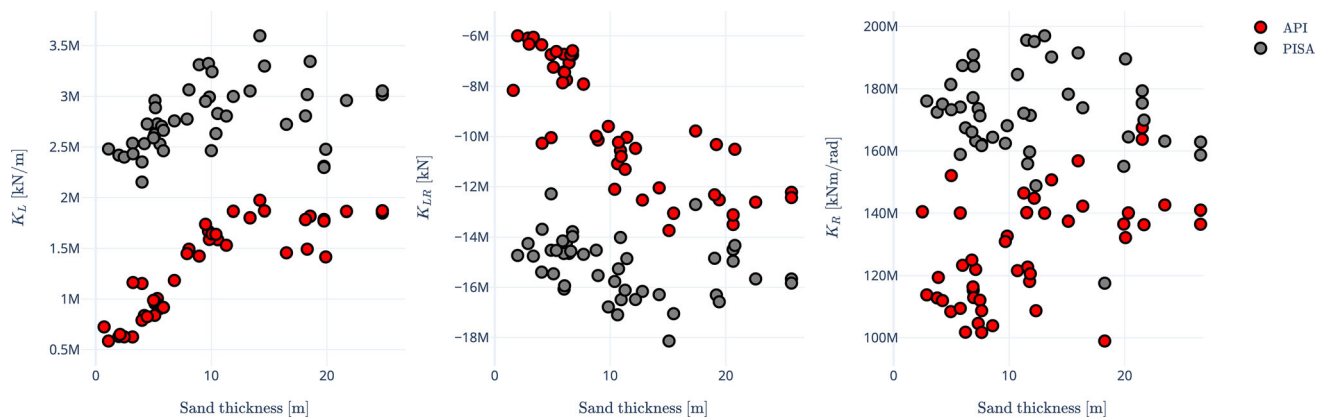


Fig. 12 Stiffness coefficients vs surface sand thickness for the entire wind farm

6 Results and discussion

The results of the farm-wide analysis calculation of stiffness coefficients based on owimetadatabase data are displayed against water depth in Fig. 11. The comparison to thickness of the surface sand layer is shown in Fig. 12. Each point represents a single foundation location.

It is immediately clear that the results confirm the conservative bias in the assessment of pile stiffness [10, 16] by the API method. The PISA methodology, established for more accurate assessment of pile-soil interaction stiffness, leads to higher values of the lateral and rotational stiffness terms. The lower off-diagonal term for the PISA method demonstrates less coupling between the lateral and rotational response. The results show a significant amount of scatter which is due to the natural variability of the soil and the variation in water depth.

However, two trends are notable from the figures. First, a correlation of the lateral stiffness coefficient K_L with the thickness of the surface sand can be identified. Indeed, a greater surface sand thickness will lead to higher stiffness. As expected, the increase in lateral stiffness with increasing

sand thickness reaches a plateau at a depth of approximately 2.5 times the monopile diameter. Around this depth, the bending moment is sufficiently absorbed by soil resistance in order not to lead to significant differences in the soil resistance mobilisation in the layers below. This suggests that the piles in the thicker sand layers are oversized.

Another exception is the rotational stiffness K_R for the PISA methodology which displays a relatively tight clustering with water depth. The rotational stiffness shows an increasing trend with water depth. There is one outlier in terms of water depth (at approximately 21m water depth) which consequently also has the lowest rotational stiffness. In general, the greatest thicknesses of surface sands are also found for the lowest water depths (due to sand waves in the wind farm area), but the variation of rotational stiffness with surface sand thickness shows much more scatter. Rather, the pile dimensions show that the deepest foundations were designed for the deepest water depths and this is believed to be a more important factor in the clustering for K_R obtained with the PISA method.

7 Conclusions and recommendations

The development of a geospatial semi-structured database for geotechnical and structural data of offshore wind foundations is demonstrated. Performing location-specific, farm-wide analysis of foundation response is only possible using a repository with reliable and accessible data. The benefits of using cloud-based solutions with a semi-structured database and associated web application for managing access to the data are highlighted. The standardised data models combined with the Application Programming Interface (API) allow access to the data by HTTP requests for authorised users.

Making data available through these standardised interfaces allows the development of parametric workflows for foundation response analysis. Both during foundation design and during the operational lifetime of the foundations, location-specific analysis allows the identification of locations with problematic soil conditions or inadequate foundation dimensions. Through the use of unstructured data fields (JSON fields), geotechnical test results from a wide variety of in-situ and laboratory tests can be included.

An example of farm-wide analysis of foundation stiffness at the mudline is used to illustrate the capabilities of the database. These results can be used further for an assessment of the natural frequencies of offshore wind monopile foundations. Because the workflow is fully parametric, sensitivity studies on the input parameters to the foundation response analyses can easily be performed.

8 API demo

The owimetadatabase API can be used on request under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. A user guide for getting started with using the API is available on GitHub (https://github.com/OWI-Lab/owimetadatabase_demo).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- American Petroleum Institute (2011) API RP 2GEO Geotechnical and Foundation Design Considerations. Tech rep, standard
- Arany L, Bhattacharya S, Macdonald JH et al (2016) Closed form solution of eigen frequency of monopile supported offshore wind turbines in deeper waters incorporating stiffness of substructure and ssi. *Soil Dyn Earthq Eng* 83:18–32
- Asiminidis C, Kokkonis G, Kontogiannis S (2019) Managing iot data using relational schema and json fields a comparative study. *IOSR J Comput Eng (JCE)* 20(3)
- Association of Geotechnical and Geoenvironmental Specialists (2017) Electronic Transfer of Geotechnical and Geoenvironmental Data. Tech Rep. Edition 4.0.4
- Bachus R, Machairas N, Cadden A (2020) Standardizing Geodata Transfer and Storage. *Deep Foundations (May/June 2020)*
- Baldini I, Castro P, Chang K, et al (2017) Serverless computing: current trends and open problems. In: *Research Advances in Cloud Computing*. Springer, pp 1–20
- Bhagal J, Choksi I (2015) Handling big data using nosql. In: *2015 IEEE 29th international conference on advanced information networking and applications workshops*, IEEE, pp 393–398
- Butler H, Daly M, Doyle A et al (2016) The geojson format. Internet Engineering Task Force (IETF)
- Byrne B, McAdam RA, Burd H, et al (2017) Pisa: new design methods for offshore wind turbine monopiles. In: *Offshore site investigation geotechnics 8th international conference proceeding*, Society for Underwater Technology, pp 142–161
- Devriendt C, Weijtjens W, El-Kafafy M et al (2014) Monitoring resonant frequencies and damping values of an offshore wind turbine in parked conditions. *IET Renew Power Gener* 8(4):433–441. <https://doi.org/10.1049/iet-rpg.2013.0229>
- Fielding RT, Taylor RN, Erenkrantz JR et al (2017) Reflections on the rest architectural style and “principled design of the modern web architecture” (impact paper award). In: *Proceedings of the 2017 11th joint meeting on foundations of software engineering*, pp 4–14
- Forcier J, Bissex P, Chun WJ (2008) Python web development with Django. Addison-Wesley Professional, Boston
- Igoe D, Fitzgerald B, Sarkar S (2019) Monopile soil-structure interaction for estimating the dynamic response of an offshore wind turbine. Tech. rep
- Jamiolkowski M, Lo Presti D, Manassero M (2003) Evaluation of relative density and shear strength of sands from cpt and dmt. In: *Soil behavior and soft ground construction*, pp 201–238
- Jung MG, Youn SA, Bae J et al (2015) A study on data input and output performance comparison of mongodb and postgresql in the big data environment. In: *2015 8th international conference on database theory and application (DTA)*, IEEE, pp 14–17
- Kallehave D, LeBlanc C, Liingaard M (2012) Modification of the API P-Y formulation of initial stiffness of sand. In: *Integrated Geotechnologies - Present and Future*, London, pp 465–472
- Kriegel A (2011) *Discovering SQL: a hands-on guide for beginners*. Wiley, Hoboken
- Kunapo J, Dasari GR, Phoon KK et al (2005) Development of a web-gis based geotechnical information system. *J Comput Civ Eng* 19(3):323–327
- Obe R, Hsu L (2011) Postgis in action. *GEOInformatics* 14(8):30
- Petković D (2017) Json integration in relational database systems. *Int J Comput Appl* 168(5):14–19
- Robertson PK (2009) Interpretation of cone penetration tests—a unified approach. *Can Geotech J* 46(11):1337–1355
- Schafhirt S, Page A, Eiksund GR, et al (2016) Influence of soil parameters on the fatigue lifetime of offshore wind turbines with monopile support structure. *Energy Procedia* p 10

23. Seidel M, Voormeeren S, van der Steen JB (2016) State-of-the-art design processes for offshorewind turbine support structures. Published in: *Stahlbau*, Volume 85, Issue 9 (9/2016) Copyright 2016 Ernst & Sohn Verlag für Architektur und technische Wissenschaften GmbH & Co KG, Berlin p 14
24. Simons B et al (2020) The new zealand geotechnical database: a national resource. The University of Auckland, Auckland
25. Stonebraker M, Rowe LA, Hirohama M (1990) The implementation of postgres. *IEEE Trans Knowl Data Eng* 2(1):125–142
26. Van Damme M, Boel K, De Nil K et al (2006) De ontwikkeling en evolutie van de databank ondergrond vlaanderen(dov): een uitwisselingsplatform voor ondergrond-gerelateerde gegevens in het algemeen en voor grondwateronderzoek in het bijzonder. In: *Congres Watersysteemkennis*
27. Weijtjens W, Verbelen T, De Sitter G et al (2016) Foundation structural health monitoring of an offshore wind turbine—a full-scale case study. *Struct Health Monitor An Int J* 15(4):389–402. <https://doi.org/10.1177/1475921715586624>
28. WindEurope (2020) Offshore Wind in Europe: Key trends and statistics 2019. Tech. rep
29. Zhang L, Yi J (2010) Management methods of spatial data based on postgis. In: 2010 Second Pacific-Asia conference on circuits, communications and system, IEEE, pp 410–413

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.