BRAINTEASER Architecture for Integration of AI Models and Interactive Tools for Amyotrophic Lateral Sclerosis (ALS) and Multiple Sclerosis (MS) Progression Prediction and Management

Vladimir Urošević[1(✉)], Nikola Vojičić[1], Aleksandar Jovanović[1], Borko Kostić[1], Sergio Gonzalez-Martinez[2], María Fernanda Cabrera-Umpiérrez[2], Manuel Ottaviano[2], Luca Cossu[3], Andrea Facchinetti[3], and Giacomo Cappon[3]

[1] Research and Development Department, Belit d.o.o. (Ltd.), Trg Nikole Pašića 9, 11000 Belgrade, Serbia
vladimir.urosevic@belit.co.rs

[2] Life Supporting Technologies Group, Universidad Politecnica de Madrid, Avenue Complutense 30, 28040 Madrid, Spain

[3] Department of Information Engineering, Università Degli Studi di Padova, Via Gradenigo 6/b, 35131 Padova, Italy

Abstract. The presented platform architecture and deployed implementation in real-life clinical and home care settings on four Amyotrophic Lateral Sclerosis (ALS) and Multiple Sclerosis (MS) study sites, integrates the novel working tools for improved disease management with the initial releases of the AI models for disease monitoring. The described robust industry-standard scalable platform is to be a referent example of the integration approach based on loose coupling APIs and industry open standard human-readable and language-independent interface specifications, and its successful baseline implementation for further upcoming releases of additional and more advanced AI models and supporting pipelines (such as for ALS and MS progression prediction, patient stratification, and ambiental exposure modelling) in the following development.

Keywords: amyotrophic lateral sclerosis · ALS · multiple sclerosis · MS · clinical disability function · ALSFRS · extended disability status · EDSS · REST services · architecture · AI models · disease progression · prediction · patient stratification · relapse · Big Data · pre-processing · regularization · daily living activities · exploratory analytics · neurodegenerative · wearable sensors · IoT integration

1 Introduction and Scientific/Technological Background

The BRAINTEASER Project (**BR**inging **A**rtificial **INTE**lligence home for a better care of **A**myotrophic lateral **S**clerosis and multiple scl**ER**osis), funded from the European Commission Horizon 2020 programme grant until the end of 2024, integrates heterogeneous societal, environmental, health, and lifestyle/habitual data from diverse sources,

K. Jongbae et al. (Eds.): ICOST 2023, LNCS 14237, pp. 16–25, 2023.
https://doi.org/10.1007/978-3-031-43950-6_2

developing patient stratification and disease progression AI models and applicative tools for improving disease management and ubiquitous monitoring and care delivery for ALS and MS patients (and assistance to informal caregivers).

Both are very complex chronic progressively degenerative fatal neurological diseases significantly disrupting the quality of life of the patients and their families, with notable differences in clinical picture, evolution, prognosis and therapies, but also many similarities in modelling and care/intervention delivery contexts (both in clinical and outpatient settings).

The initial releases of the novel interactive applicative tools for disease management and monitoring, currently in use in real-life settings in four clinical study validation sites (Lisbon, Madrid, Pavia and Turin) of the Project, are described with main features and functionalities in [1], including also some of the basics of the underlying back-end platform architecture and implementation. These tools and platform middleware services are enabling and supporting the first crucial step in the main overall identified and modelled BRAINTEASER data and process flows – continuous acquisition, ingestion, integration and storage of:

- detailed retrospective and prospective clinical datasets, with the prospective ones collected in the course of the Project studies being additionally complemented and augmented by
- comprehensive heterogeneous personal health, activity, lifestyle, habitual/behavioural, and environmental data, collected using:
  - digitalized instruments and questionnaires for ALS and MS (and comorbidities) clinical evaluation and remote disease progress assessment, both standardized and in common practice (like ALSFRS-R or EDSS), as well as some innovative and evolving ones (like Awaji-Shima Consensus or Gold Coast diagnostic criteria for ALS), and
  - commonly available sensing/IoT devices, mainly Garmin smartwatches, and portable and fixed air quality and atmospheric para-meters sensing devices (like Atmotube PRO[1] and PurpleAir (Classic) PA-II[2]).

The collected data have driven the development of Artificial Intelligence (AI) models able to address the needs of precision medicine, enabling early risk prediction of disease fast progression and adverse events. During the previous yearly period the intense development and evaluation of AI models in the Project (including relevant Open Science efforts co-organized by the Project[3,4] [2]) have further resulted with the first releases of the model routines, tested and delivered ready for integration.

These generate not only the main envisioned final model outputs - such as predictions of probabilities or timeframes of occurrence of key disease progression events, like MS relapse or introduction of NIMV (Non-Invasive Mechanical Ventilation) or PEG (Percutaneous Endoscopic Gastrostomy) treatment for a patient - but also include

[1] https://atmotube.com/atmotube-pro.

[2] https://www2.purpleair.com/products/purpleair-pa-ii.

[3] https://brainteaser.health/open-evaluation-challenges.

[4] https://brainteaser.health/open-evaluation-challenges/idpp-2022.

the pre-processing or re-calibration model routines within the models, generating intermediate summary aggregated or transformed values from raw data inputs fetched from the unified platform Data Store, and passing the results back again to be persisted in the Data Store and re-used mostly for provision to the consuming applications exposed to the targeted end users, and as inputs to other following routines further in the disease progression prediction and patient stratification model pipelines. Such pre-processing routines, having been released first as initial steps in the ad-hoc AI processing pipelines, are actually turning out to be the most demanding ones from the integration perspective, as they require more frequent periodic invocation and execution (up to several times daily), and generate much higher throughput and consumption of data exchanged with the core platform services than the routines for actual disease progression prediction and patient stratification (invoked once in weeks or even months, and relying on already pre-processed results as more compacted inputs). Main algorithm types exploited in these pre-processing, feature extraction/selection and dimensionality reduction routines are Bayesian filtering and smoothing, retiming, or oximetry digital biomarkers evaluation (*pobm*[5] package), while the further AI models for continuous disease monitoring and progression prediction yet to be deployed exploit survival analysis algorithms like Cox proportional hazards, supported by methods like forward-recursive feature selection, and others elaborated fully and in detail in related publications like [10].

This paper describes architecture and implementation of the core BRAINTEASER platform back end and services tier integrating and supporting the mentioned released AI model routines in operation (as well as other recently developed or advanced supported features of the data feeding and consuming applications or modules coupled to the platform, presented summarily as a reminder on the general ecosystem and flows overview on Fig. 1 below). The general integration approach, as described in the following sections, is the same for both abovementioned types of AI model routines for now, with eventual specific critically performance-dependent alternative pathways and design patterns supporting tighter coupling or more extensive query stream parallelism having also been developed in reserve, and used for some scenarios of environmental/ambiental data processing, as described in [8]. Pilot demonstration and validation phase of the Project (initially focused on ubiquitous personal data collection, cleaning and integration at this stage, as mentioned) has just started a couple of months ago at the time of writing of this paper, with still a limited number of recruited study subjects and collected feedback and data on the usage. After at least a further semester of increased recruitment and more intensive continuous usage of the BRAINTEASER platform applications deployed towards the end-users, there will hopefully be sufficient data and statistics on the usage of the platform for at least an elementary sound and substantial analysis of the overall results and performance of the deployed implementation of the architecture to complement and expand on the work reported here, requiring more space than available in this short conference publication format, possibly in an evolved derived journal article.

[5] https://pypi.org/project/pobm.

2 Architectural Overview

The diagram on Fig. 1 below provides a broad overview of the overall envisaged BRAINTEASER Data and ICT Tools ecosystem with main data flows and structural breakdown of data consuming or sourcing components and modules (more detailed in particular of the services tier in the biggest frame top right, crucial for integration), and the logo icons next to main modules/tiers to be described additionally denoting the system infrastructure language or platform stack chosen for development and deployment (- Python, - PostgreSQL, - Java).

The presented schema is an evolution and expansion of the similar initial one provided as Fig. 7 in [1], and as noted in that referenced article, some of the platform functionalities and architecture build and extend upon the related efforts and outputs from preceding and parallel related projects, specifically from PULSE[6] (partially described in [3, 5, 7]), NEVERMIND[7] (in [6]), and PERISCOPE[8] (in [4]).

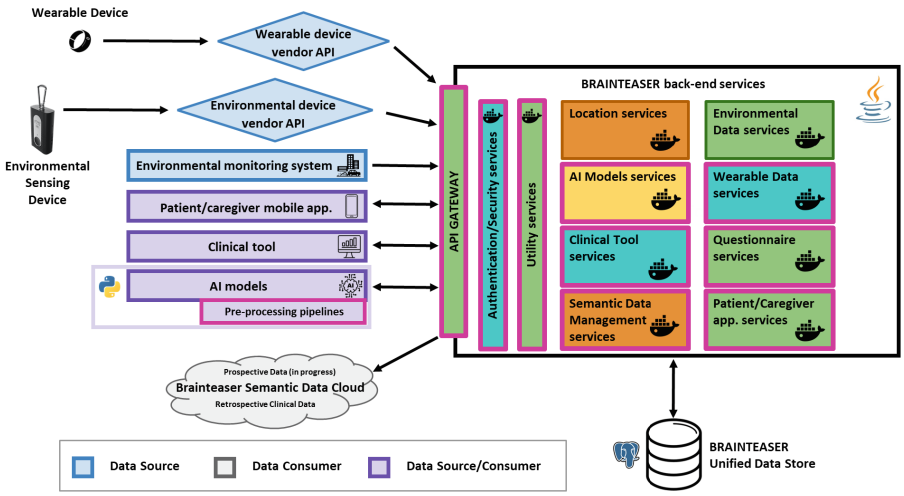

Fig. 1. Updated BRAINTEASER ecosystem architecture with main data sources, consumers and flows, and detailed service tier breakdown.

RESTful APIs are the main interfacing and architectural approach on the back-end tier, with secure communication between the web services enhanced by industry standard JWT[9] (JSON Web Token) lightweight encrypted encapsulation of request and response payloads. This extends also to the implementation and deployment of the AI models and routines themselves, with common unified human-readable and understandable interfaces specification based on JSONs, agnostic of the language the wrapped underlying

[6] https://www.project-pulse.eu.
[7] https://www.nevermindproject.org.
[8] https://periscopeproject.eu.
[9] https://jwt.io.

logic is written in, being strongly preferred across the complete services and tools ecosystem. Some of the additional key benefits of this for facilitated maintenance, scalability and sustainability of the platform beyond the Project developments are:

- OpenAPI (3.x) Specification (OAS) compliance, with ample open and available support and standardized toolsets for easier and semi-automated generation and maintenance of API specification and documentation live online (currently on Swagger[10] at https://brainteaser.belit.co.rs/gateway/swagger-ui/index.html, with relevant example provided on Fig. 2 below), as well as for API testing, mocking, and overall lifecycle governance and scaling.

  Collections feature[11] of the Postman REST API platform client has also been extensively used for building and maintaining request sets for practical integration testing of intra-service communication and data flows, and scheduled periodic batch server jobs, needed for execution management of most AI model routines, are also easier to consolidate and manage with uniform service calls (though still specific to application server/container or hosting server OS).

- the encapsulation of AI model and preprocessing routines (commonly written by data scientists as Python or R functions) within standard web service endpoints, similar to the interfacing (Java-based) invoking and managing ones of the core platform, also unifies and simplifies the deployment and CI/CD (Continuous Integration/Continuous Delivery) pipelines across the platform, including deployment containerization and scaling (with tools like Docker).

  Concrete specifics of this web application server/container "wrapping" implementation for Python model routines are provided in the following section.

Development practice at this stage is to have each specific thematic domain set of AI models (for disease monitoring, progression prediction, patient stratification…) encapsulated in a dedicated wrapping microservice as it gets completed and delivered for platform integration (Fig. 3). Later towards the release of the overall integrated ecosystem, refactoring for optimal modularity can be performed, possibly merging some of the microservices (or most of them, into a practical modular monolith architecture), according to the results of the continuous models screening, in-silico simulation, evaluation, and improvement pipeline in the scope of the Project WP4, and according to the finally identified performance requirements and constraints.

The overall back-end service tier is similarly structured, mainly based on loosely coupled REST microservices [9], but with some practical trade-offs towards modular monolith, and the separation between domain-specific and infrastructural/utility orthogonal logic. The service packages and sets specific to the BRAINTEASER main thematic domains logic (supporting features and functionalities related to patients, caregivers, diseases & comorbidities, IoT devices and data in the system, etc.), presented as horizontal rectangles in the main top right frame on Fig. 1, are grouped into a couple of subprojects in development, and are currently using just three separate schemas/tablespaces in the Data Store underneath, divided at this stage mainly according to non-functional

[10] https://swagger.io.

[11] https://www.postman.com/collection.

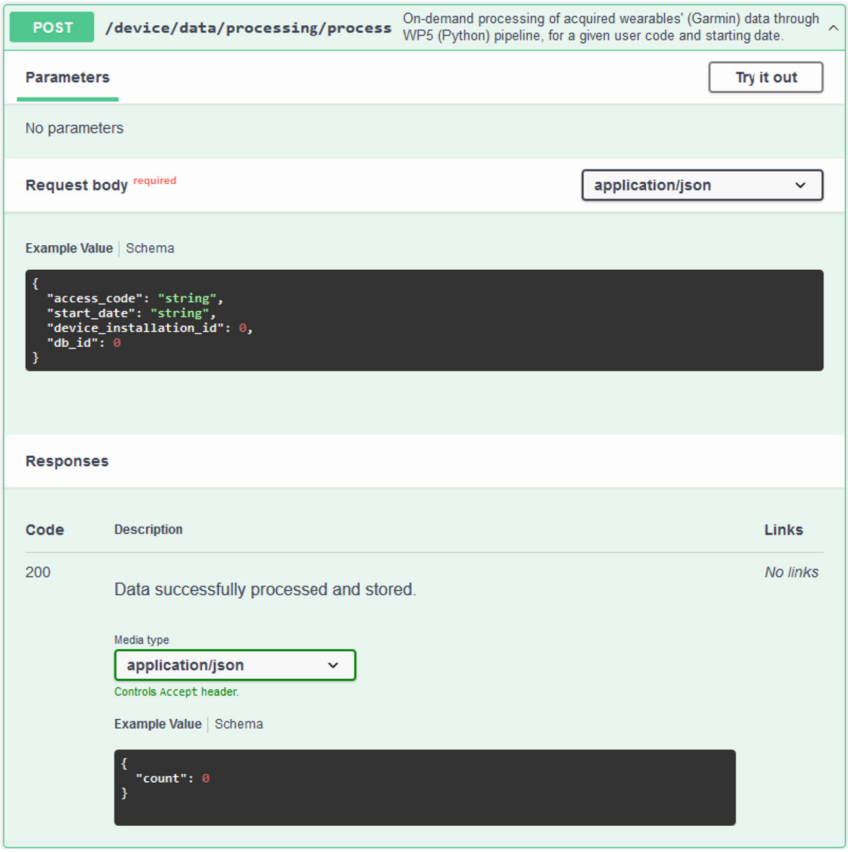

Fig. 2. Example of testable specification/documentation of a platform API endpoint for on-demand invocation of a data pre-processing method generating inputs for the AI models for continuum disease monitoring (developed within the Work Package 5 (WP5) of the Project).

requirements – one containing highly sensitive potentially personally-identifying data (descriptive reported symptoms, specific socio-demographic and profile data…) kept protected fully encrypted in the database and throughout all handling in the system, other with the "regular" non-protected or already de-identified data, and the third for metadata. As the ingested data volume increases, the first two will likely separate to at least another additional dedicated to most bulky IoT sensed measurements data.

Microservices with separate underlying schemas supporting the infrastructural functionalities orthogonal to the domain logic (access control, authentication, security, utilities…) are represented as rectangles with vertical labels on Fig. 1, with the API Gateway package implementing the design pattern for basic service orchestration. Subsystems are hidden behind the Gateway façade service, acting not only as a proxy to those domain services but also validating requests (in terms of tokens, basic structure, sequencing…)

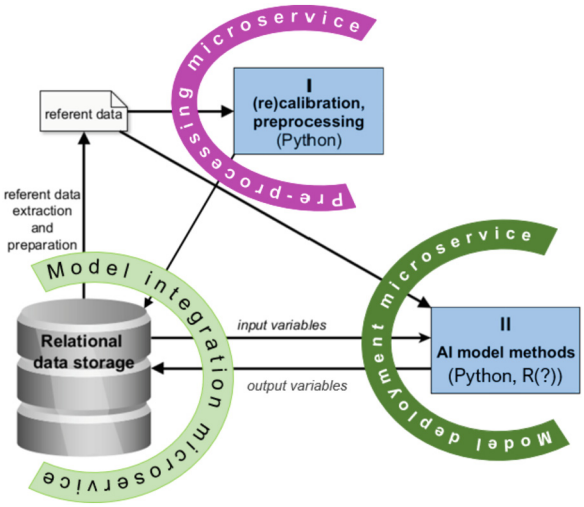

Fig. 3. Generic pipeline flows and microservice encapsulations of the integration of AI model and pre-processing methods with the core BRAINTEASER Platform services and Data Store

and documenting the specifications of all services through Swagger. Gateway also implements composition of calls for operations requiring calls to multiple services (or multiple calls to a single service), which is preferred to complex network of direct calls between sub-services. Code for request and response model, and endpoint definitions (URI, method, and docs) of sub-services are shared with Gateway by Git submodules in CI/CD.

Some additional most prominent employed service design patterns, mainly for the data collection, fusion, and provision to the applicative tools for disease monitoring and management, are described in [8].

3 Implementation

Service tier core is currently implemented on the robust heavy-duty industry standard and proven Enterprise Java (Amazon Coretto 17 LTS[12]) web technology stack, leveraging Spring Boot[13] framework 2.6.4 with all the functional programming and REST APIs support. Upgrade to the next Java LTS (Long-Term Support) version, expected to be Java 21 released around September 2023, is planned and being prepared for jOOQ[14] (Java Object Oriented Querying) framework is used for object-relational mapping (ORM), and all is deployed and running on the Apache Tomcat 9.x web application server.

The industry-standard and common jUnit 5 framework[15] is used for unit test generation, with REST Assured[16] for baseline Java REST services semi-automated testing

[12] https://docs.aws.amazon.com/corretto/latest/corretto-17-ug/what-is-corretto-17.html.
[13] https://spring.io/projects/spring-boot.
[14] https://www.jooq.org.
[15] https://junit.org/junit5.
[16] https://rest-assured.io.

purposes, along with other extensive REST API development and lifecycle management tools specified above.

Other common and mostly newer open-source alternatives, like for example a complete Python-based stack covering also the AI models implementation, or JavaScript server-side implementations, still do not offer as comprehensive and robust support and performance in REST API services implementation, both object-oriented and functional programming, and cross-platform (heavy-duty web and mobile applications being the key BRAINTEASER tools for the end users) development.

GitLab serves as the code repository and version control system, as well as for CI/CD pipelines and control (except for the data tier, where Red Gate Flyway[17] tool is used for database versioning management and migration/replication control).

Parallel Python (3.x.x) runtime is hosting the integrated AI model methods for sensory data pre-processing and disease monitoring, encapsulated in web service pipelines as described above, using the RESTful API implementation via the lightweight Flask[18] web framework deployed along with the Java-based core services stack on the platform back-end server cloud, with SQL Alchemy[19] being the optional ORM framework in the Python-based stack. Principally, similar language-agnostic implementation would be viable for all AI models envisioned to be developed in the Project (more details in the next section with conclusions), supporting loosely coupled APIs for seamless scaling or significant changes (in cases like e.g., specific model routines getting completely rewritten and implemented in R or Julia, REST-based interfacing and service invocation and deployment control should remain unchanged, if a corresponding frameworks like Plumber or Genie are used instead of Flask).

Unified JSON-based communication across the platform also provides for some advantages on the data tier – implemented as PostgreSQL hybrid-relational Data Store, it features extensive support for JSON and binary JSON data types, querying, indexing and optimization. Consequently a lot of data that are natively structured in JSON documents as collected or generated in the ecosystem (evolving generic questionnaires, service configurations, intervention and gamified content…) are stored and queried in JSON format in the database, and fully deserialized into relational model entities only when necessary for main data consumption and performance criteria, or the structure evidently standardized and fixed in the long-term or permanently (like for the standardized questionnaire instruments exemplified in Sect. 1 above, or data model structures compliant with the relevant architectural standards in healthcare IT, mainly ISO/CEN 13606, openEHR, and HL7 FHIR…). This implementation has in development and deployment practice experiences by now shown equal or comparative performance in handling the mentioned document-structured data as using dedicated document-oriented databases like MongoDB, while at the same time retaining advantages of native relational data support (most of the data handled by the overall platform are relational by nature) or more comprehensive and robust transaction control (nesting, cascading), all in a single unified data store managing the complete heterogeneity of data. PostgreSQL has also shown satisfactory overall performance with the terabyte-level volumes of sensed

[17] https://flywaydb.org.

[18] https://palletsprojects.com/p/flask.

[19] https://www.sqlalchemy.org.

IoT data (billion-record tables) expected to be collected by the end of the Project, with proper indexing and query optimization, and using multiple available and well supported dedicated extensions (for scalability, time-series data management, etc.).

Leveraging the JSON-LD[20] format for Linked Data is also convenient over the complete uniform JSON platform interfacing and communication, for seamless expected upcoming semantic integration, requiring minimal overhead efforts and data model changes, with the BRAINTEASER Semantic Cloud [2] that has been developed and evolved during the two initial Project years mostly independently from the described platform. A referent similar semantic integration example is shortly described in Sect. 4.3 in [4].

4 Conclusion and Further Development and Evolution

The presented platform architecture and deployed implementation in real-life clinical and home care settings on four BRAINTEASER study sites, integrating the novel working tools for improved ALS and MS monitoring and management released last year with the initial releases of the AI models for disease monitoring (and the supporting data pre-processing pipeline).

This integration of two key types of targeted ICT outputs of the BRAINTEASER Project through the described robust industry-standard scalable platform is to be a referent example of the integration approach based on loose coupling APIs and industry open standard human-readable and language-independent interface specifications, and its successful baseline implementation for further upcoming releases of additional and more advanced AI models and supporting pipelines (such as for ALS and MS progression prediction, patient stratification, and ambiental exposure modelling) in development until the end of 2024.

Acknowledgement. The presented work and developments are being funded by the European Commission Horizon 2020 programme grant agreement GA101017598, in the scope of the BRAINTEASER project.

References

1. Gonzalez-Martinez, S., et al.: Novel interactive BRAINTEASER tools for amyotrophic lateral sclerosis (ALS) and multiple sclerosis (MS) management. In: Aloulou, H., Abdulrazak, B., Mokhtari, M., et al. (eds) Participative Urban Health and Healthy Aging in the Age of AI (Proceedings of ICOST 2022 conference). Lecture Notes in Computer Science, **13287**. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-09593-1_26
2. Aidos, H., et al.: iDPP@CLEF 2023: the intelligent disease progression prediction challenge. In: Kamps, J., Goeuriot, L., et al. (eds). Advances in Information Retrieval. Proceedings of the 45th European Conference on Information Retrieval (ECIR) 2023. Lecture Notes in Computer Science, **13982**. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-28241-6_57

[20] https://json-ld.org.

3. Ottaviano, M., et al.: Empowering citizens through perceptual sensing of urban environmental and health data following a participative citizen science approach. Sensors **19**(13), 2940 (2019). https://doi.org/10.3390/s19132940
4. Pala, D., et al.: A new interactive tool to visualize and analyze COVID-19 data: the PERISCOPE atlas. Int. J. Environ. Res. Public Health (IJERPH), **19**(15), 9136 (2022). https://doi.org/10.3390/ijerph19159136
5. Ottaviano, M., et al.: Participative app. for citizens to assess health risks and increase pollution awareness. In: Proceedings of the IEEE EMBS International Conference on Biomedical & Health Informatics (BHI), pp. 1–1 (2019). https://ieeexplore.ieee.org/document/8834574
6. Petros, N.G., et al.: Sociodemographic characteristics associated with an e-health system designed to reduce depressive symptoms among patients with breast or prostate cancer: prospective study. JMIR Formative Res. **6**(6), e33734 (2022). https://formative.jmir.org/2022/6/e33734
7. Vito, D., Ottaviano, M., et al.: The PULSE project: a case of use of big data towards a comprehensive health vision of city well-being. In: Jmaiel, M., Mokhtari, M., Abdulrazak, B., Aloulou, H., Kallel, S. (eds) The Impact of Digital Technologies on Public Health in Developed and Developing Countries, Proceedings of ICOST 2020 Conference, Lecture Notes in Computer Science, **12157**. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-51517-1_39
8. Urošević, V., Dagliati, A., Ottaviano, M., Vojičić, N., Larizza, C. Pala, D.: Design and optimization of REST services for performance and scalability in provision of big environmental data to exploratory analytics of their effects on progression of ALS and MS. In: Proceedings of the 2022 IEEE 12th International Conference on Consumer Electronics (ICCE-Berlin), Berlin, Germany, pp. 1–6 (2022). https://doi.org/10.1109/ICCE-Berlin56473.2022.9937100
9. Kocher, P.S.: Microservices and Containers. Addison-Wesley Professional (2018)
10. Pancotti, C., et al.: Deep learning methods to predict amyotrophic lateral sclerosis disease progression. Sci Rep. **12**(1), 13738 (2022). https://doi.org/10.1038/s41598-022-17805-9