Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 196 (2022) 615–622

Procedia Computer Science

www.elsevier.com/locate/procedia

CENTERIS - International Conference on ENTERprise Information Systems / ProjMAN - International Conference on Project MANagement / HCist - International Conference on Health and Social Care Information Systems and Technologies 2021

Design and Implementation of Medical Searching System Based on Microservices and Serverless Architectures

Jawad Sadek[a]*, Dawn Craig[a], Michael Trenell[b]

[a]*NIHR Innovation Observatory , Newcastle University, Newcastle, UK*
[b]*Faculty of Medical Sciences Newcastle University, Newcastle, UK*

Abstract

Microservices and Serverless computing are promising cloud-based architectural models in the software development industry that have many advantages over previous technology models. The benefit of adopting these more novel models, however, is dependent on the volume of the application workload and the execution behavior. ScanMedicine is a new searching system dedicated to providing health care professionals, patients and researchers with easy access to intelligence underpinning health technology innovations. We present the design and implementation of ScanMedicine's framework using AWS lambda functions and microservices. We incorporated a layered architectural style where each layer run on separate hardware and adopt different architectural technique.

Keywords: Microservices, Serverless, Cloud Computing, AWS, Clinical Trials, Medical Devices;

* Corresponding author. *E-mail address:* jawad.sadek@newcastle.ac.uk

1. Introduction

The World Health Organization (WHO) defines a clinical trial as “any research study that prospectively assigns human participants or groups of humans to one or more health-related interventions to evaluate the effects on health outcomes. Interventions include but are not restricted to drugs, cells and other biological products, surgical procedures, radiological procedures, devices, behavioral treatments, process-of-care changes, preventive care, etc.” [1]. The revised Declaration of Helsinki, widely regarded as the cornerstone document on human research ethics, states “every **clinical trial** must be **registered** in a public accessible database before recruitment of the first subject” [2]. To facilitate this there are numerous national and international registries, such as the national Chinese Clinical Trial Register (ChiCTR) or international registers such as the International Standard Randomised Controlled Trial Number (ISRCTN) and ClinicalTrial.gov [3]. Trial registries are web-based portals which provide access to information on registered clinical studies, often including protocol summaries, interim results and summary of final results [4]. Searching clinical trial registries to capture ongoing or completed, but not yet formally published, trial data is considered an essential process for horizon scanning and to support other forms of secondary research related to the assessment of clinical effectiveness. Examples of such studies are systematic reviews [5], meta-analyses [6], health technology assessments (HTA) and early awareness [7]. The World Health Organization (WHO) host and maintain the International Clinical Trials Registry Platform (ICTRP) that combines the content of a number of registers in one platform. However, ICTRP is updated by data providers who submit their trial records every week or four weeks which could result in a delay in refreshing trial information [8].

ScanMedicine is a new searching system that curates medicine data from clinical trial repositories and medical devices approved by the US Food and Drug Administration (FDA). It aims to providing health care professionals, patients and researchers with easy access to intelligence underpinning health technology innovations. The system is being developed at the National Institute for Health Research Innovation Observatory (NIHRIO); a research center hosted by the Newcastle University. The reminder of this paper is organized as follows: Section 2 discusses two architectural models in software development and describes the software architecture of ScanMedicine, Section 3 presents the underlying technology employed to implement the proposed architecture, Section 4 reviews literature related to clinical trials searching systems, and Section 5 concludes the paper with points of future enhancement. ScanMedicine can be accessed at https://www.scanmedicine.com

2. Software Architecture

2.1. Serverless and Microservices models

Organizations have started adopting microservices and serverless architectures to respond more rapidly to the never-ending, ever-changing demands of their business needs. These two architectures models were originally created as an alternative to the monolith approach. They possess similar functional properties – that is, ease of deployment, agile nature of handling development requirements and the minimum operational costs. However, there are some prominent differences when architecting software solutions.

The Microservices paradigm is an extension of service-oriented-architecture (SOA) [9] where each fine-grained service is tied to a specific function [10]. Services are loosely coupled and independent of each other with an adoption of industry standard interface/protocol such as JSON and HTML. Ideally, the implementation of the service is completely hidden and encapsulated through a containerization platform like Docker platform. This provides developers a full access to relevant libraries hosted on container whilst deploying various functional modules. The trade-off of this flexibility is the added complexity. Managing and monitoring a multitude of distributed services at scale is troublesome. Also deploying microservice solution on the cloud may turn out to be pricey since microservices is a request-response driven tech solution i.e., services are always active even when sitting idle. Therefore, business owners have to pay for host servers’ capacities on monthly/yearly basis.

On the other hand, serverless is an event-driven type of architecture allowing application developers to run custom codes in stateless compute containers [11]. Serverless applications are often modelled as a collection of stateless functions that triggers cloud-based working environment in response to events. Functions (also referred as

Functions as a Service “FaaS”) are custom code run in ephemeral containers managed by a third-party cloud provider. AWS Lambda [12], Azure Functions [13] and Google Cloud Functions [14] are some popular examples of FaaS platforms. Such an architecture reduces operational costs, complexity, and operational management overheads. Applications get auto-scalability and high availability without additional effort from the developer. It gives benefit over microservices when cost and engineering time is an issue since it removes the need for a traditional always run on server components. However, giving backend services control to platform providers has some limitations. For instance, error analyses become more complicated since developers do not have visibility into the system processes, further in-depth performance and monitoring is restricted to whatever the vendor provides, and start-up latency needed to load the function especially JVM-implemented functions and those require connections to RDBMS and the run time duration imposed by FaaS provider. Unfortunately, one architectural solution rarely fits business functions and computing needs perfectly. To leverage the benefits from both Serverless and Microservices architectures, we have incorporated a layered architectural style where each layer run on a separate hardware and adopt different architectural technique.

2.2. Proposed Architecture

In architecting our software, we tried to find a solution that are vendor agnostic i.e., without using AWS Glu or AWS Batch for example. However, in the implementation we have relied on AWS. Our System architecture is composed of three layers as illustrated in Fig 1: Core (System) services layer, Integration services layer and API services layer. The goal of layering is to increase flexibility while also structuring the architecture in a way that provides abstraction and separation of concerns. Each layer run on separate hardware and has its own access control ensuring its services are protected and only a specific neighboring layer has access.

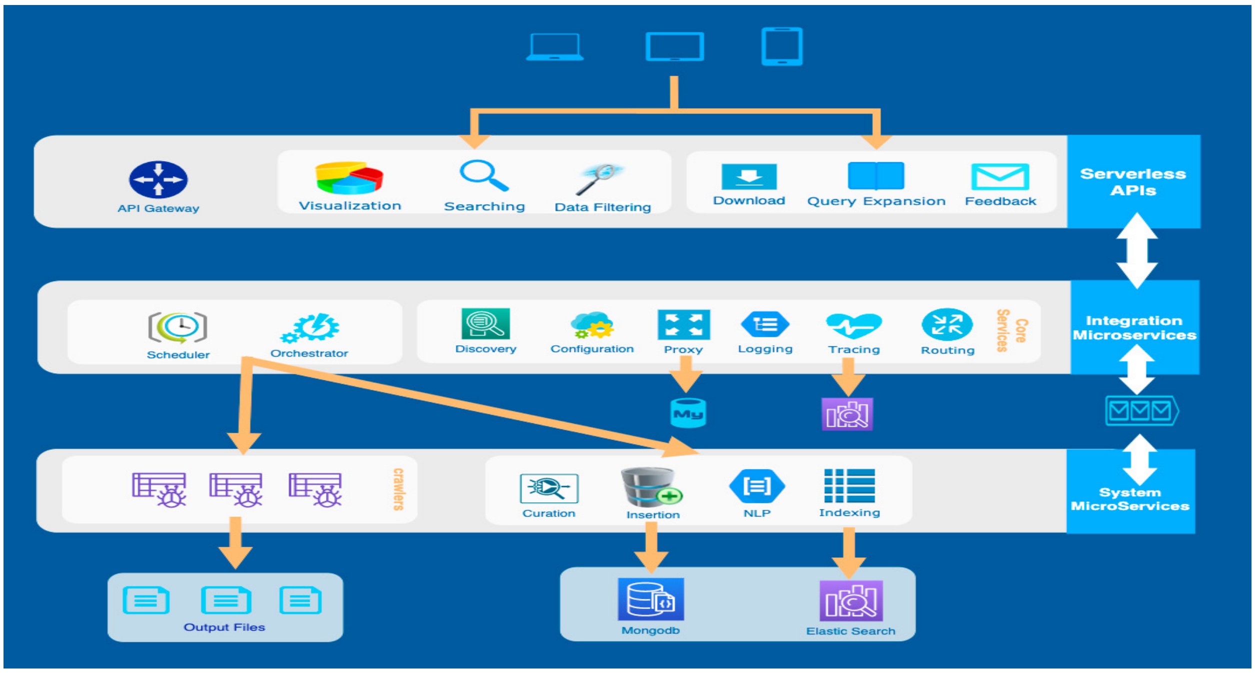


Fig. 1. High-level architecture diagram of ScanMedicine

- **Core Layer:** At the bottom layer, we have fine-grained self-contained services. Each service is discrete and encapsulates a set of responsibilities. This layer hosts the core processes, it comprises data manipulation tasks and most of the batch processing jobs including web crawling services, data extraction pipelines, natural language processing and Indexing tasks. We adopted microservices model for services located in this layer. As we discussed in the previous section, serverless functions are not intended for long-running processes due to execution time limit imposed by providers; for example, AWS Lambda functions are at present aborted if they run for longer than 15 minutes. Therefore, keeping a web-socket connection open for a web crawler is not possible. Services that are responsible to persist data into the database tier are also hosted in this layer.

- **Integration Layer:** In the middle layer, multiple services from the Core layer are linked together to form a specific functionality. In this layer, we run daily/monthly automated jobs using scheduled pipelines and workflow orchestration services. In addition, this layer accommodates other more coarse-grained services which compose the microservices ecosystem. These services are responsible to integrate and regulate network communication logic between services for example, service discovery, central configuration management, dynamic routing, load balancing, monitoring, request tracing. Also, we set up a message broker which enables asynchronous communication between services even if there were written in different languages or implemented in different platforms. The use of these solutions gives applications scalability, adaptability and improves fault tolerance.
- **Experience/Edge Layer:** At the top layer, we expose the main system capabilities using the serverless approach. The purpose of services running in this layer is to interact with end users such as Search Queries, Data Filtering, Term Expansion and Visualization. Since these services tend to be short-running tasks which are intended to return response to request rapidly, serverless functions are a great time and money saving solution. Functions within the space of this layer are stateless and unidirectional. If a function errors, then it only affects that function rather than the rest of the application. API gateway is also running as the single-entry point into the system proxying communications between serverless services. This ensure policies, like security patterns, rate limiting, and throttling are correctly applied across all functions.

3. System Implementation

Our system is hosted on AWS platform. Figure 2 presents the main infrastructure components we used to deploy ScanMedicine. Hashing and other output files produced by the web crawlers are stored in Amazon S3 buckets. Results from the data processing pipelines are consolidated and persisted into MongoDB and MySQL databases, then are indexed into ElasticSearch. High data availability and ease of access in the event of unexpected errors such as system crash are achieved through data replication and sharding. A Replica set of three MongoDB instances is deployed across private subnets of a Virtual Private Cloud (VPC). A cluster of Elasticsearch nodes is placed within these subnets. Also, MySQL master and standby servers are deployed in the same VPC.

Microservices run in the Core Layer are coded using Java, Python and NodeJs languages which cover the functionality shown in Fig 1. Services hosted in the Integration Layer are developed using Spring Cloud framework. This framework facilitates building the core components and skeleton architecture that are responsible to integrate all services into the microservices architecture for example: Service Discovery, Central Configuration and Dynamic Routing. We used Apache Maven as a project management tool and RabbitMQ as a message broker. Microservices are encapsulated in Docker containers based on Linux OS and hosted in ECS servers which are also distributed across the private subnets. Autoscaling is enabled to provide high applications availability. Elasticsearch-Logstash-Kibana (ELK) stack is integrated as a log management and analytics solution to monitor process and service logs. The stack gains valuable insight on failure diagnosis and services performance. Table 1 shows the technology stacks used to implement the middle layer.

Table 1. Technology stack used in the integration layer.

Service	Tool	License
Message broker	RabbitMq	Mozilla
Dynamic Routing	Zuul	Netflix
Service Discovery	Eureka	Netflix
Continuous Integration	Jenkins	MIT
Request Tracing	Zipkin	Apache
Containerization	Docker	Apache

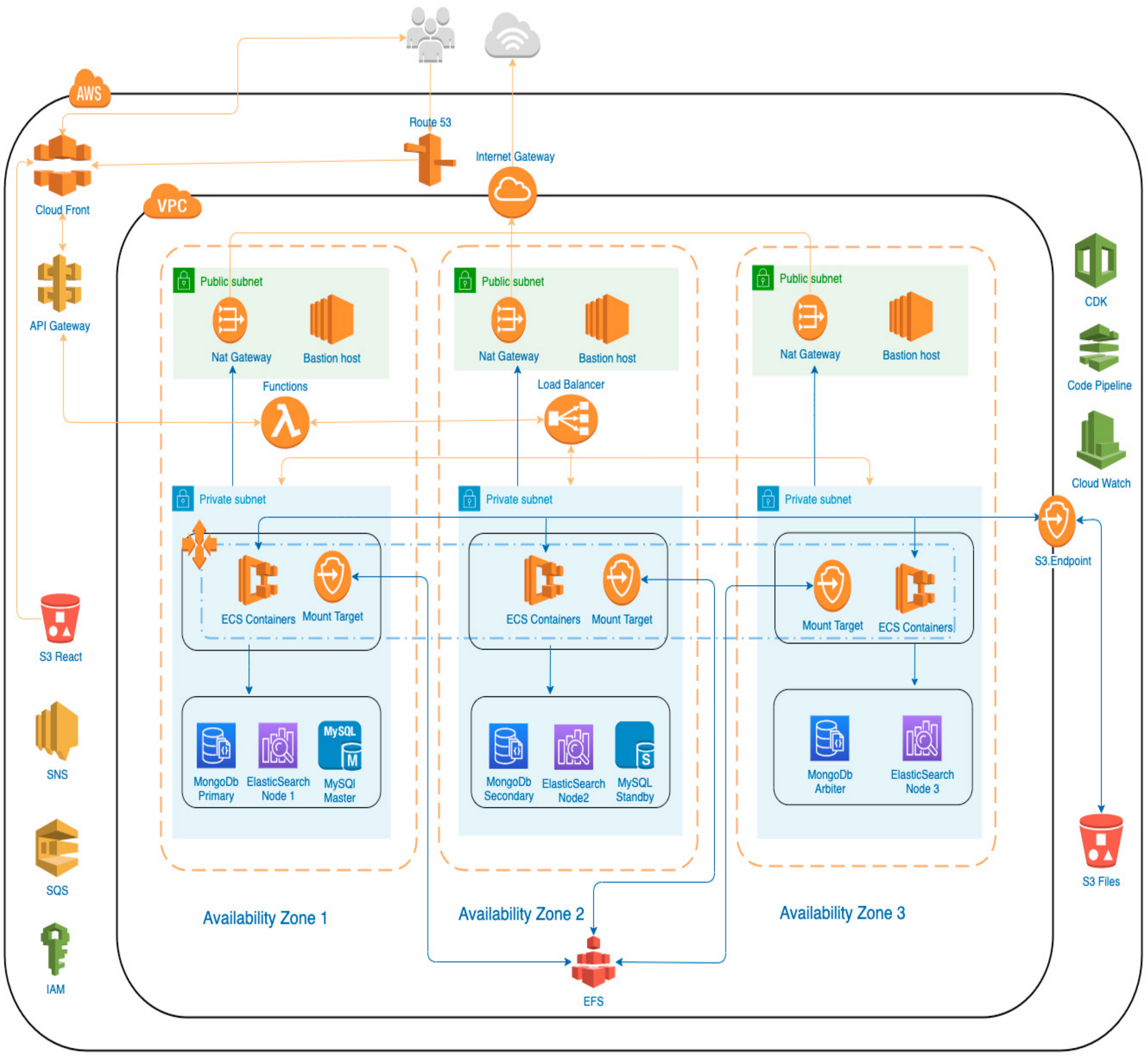
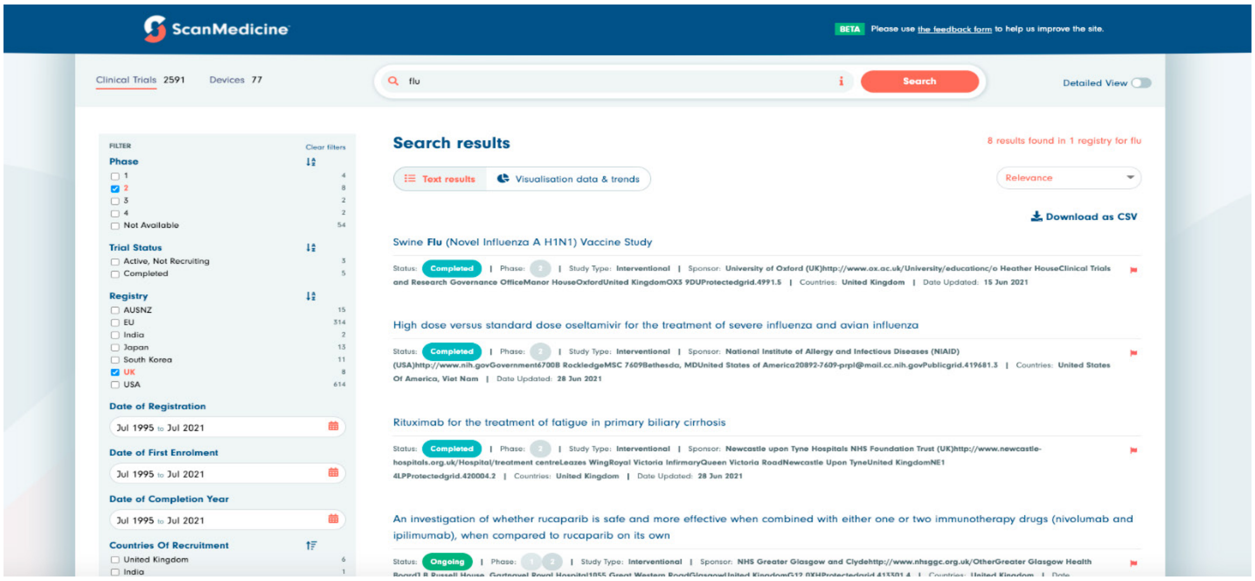


Fig. 2. System Infrastructure on AWS

The front-end is implemented with React. The user interfaces components are served through CloudFront CDN which is pointing to S3 object buckets. The source code is stored in AWS CodeCommit. The front end is built and tested from there using AWS CodeBuild server and later deployed to the bucket with a help of AWS CodePipeline. User requests are connected to the working logic through Amazon API gateway which routes the requests to the Lambda functions.

The homepage for ScanMedicine (phase one) is presented in Fig. 3. The user inserts a query in the search box and selects a dataset (clinical trials or FDA devices). The query is expanded with a list of medical synonyms obtained from the Unified Medical Language Systems (UMLS) which are presented under the search box. Users can exclude the synonyms by adding [only] to the end of the query. Users can also broaden or narrow the scope of their research using logical operators such as AND, OR, NOT. The search results are presented in the same page as an overview of all relevant clinical trials or FDA devices, each provides a direct link to the corresponding data source.


ScanMedicine
BETA Please use the feedback form to help us improve the site.
Clinical Trials 2591
Devices 77
Search
Detailed View

flu

FILTER Clear filters

Phase
- [ ] 1 4
- [x] 2 8
- [ ] 3 2
- [ ] 4 2
- [ ] Not Available 54

Trial Status
- [ ] Active, Not Recruiting 3
- [ ] Completed 5

Registry
- [ ] AUSNZ 15
- [ ] EU 314
- [ ] India 2
- [ ] Japan 13
- [ ] South Korea 11
- [x] UK 8
- [ ] USA 614

Date of Registration: Jul 1995 to Jul 2021

Date of First Enrolment: Jul 1995 to Jul 2021

Date of Completion Year: Jul 1995 to Jul 2021

Countries Of Recruitment
- [ ] United Kingdom 6
- [ ] India 1

Search results

8 results found in 1 registry for flu

Text results | Visualisation data & trends

Relevance

Download as CSV

Swine **Flu** (Novel Influenza A H1N1) Vaccine Study

Status: Completed | Phase: 2 | Study Type: Interventional | Sponsor: University of Oxford (UK)http://www.ox.ac.uk/University/educationc/o Heather HouseClinical Trials and Research Governance OfficeManor HouseOxfordUnited KingdomOX3 9DUProtectedgrid.4991.5 | Countries: United Kingdom | Date Updated: 15 Jun 2021

High dose versus standard dose oseltamivir for the treatment of severe influenza and avian influenza

Status: Completed | Phase: 2 | Study Type: Interventional | Sponsor: National Institute of Allergy and Infectious Diseases (NIAID) (USA)http://www.nih.govGovernment6700B RockledgeMSC 7609Bethesda, MDUnited States of America20892-7609-prpl@mail.cc.nih.govPublicgrid.419681.3 | Countries: United States Of America, Viet Nam | Date Updated: 28 Jun 2021

Rituximab for the treatment of fatigue in primary biliary cirrhosis

Status: Completed | Phase: 2 | Study Type: Interventional | Sponsor: Newcastle upon Tyne Hospitals NHS Foundation Trust (UK)http://www.newcastle-hospitals.org.uk/Hospital/treatment centreLeazes WingRoyal Victoria InfirmaryQueen Victoria RoadNewcastle Upon TyneUnited KingdomNE1 4LPProtectedgrid.420004.2 | Countries: United Kingdom | Date Updated: 28 Jun 2021

An investigation of whether rucaparib is safe and more effective when combined with either one or two immunotherapy drugs (nivolumab and ipilimumab), when compared to rucaparib on its own

Status: Ongoing | Phase: 1 2 | Study Type: Interventional | Sponsor: NHS Greater Glasgow and Clydehttp://www.nhsggc.org.uk/OtherGreater Glasgow Health Board1 B Russell House, Gartnavel Royal Hospital1055 Great Western RoadGlasgowUnited KingdomG12 0XHProtectedgrid.413301.4 | Countries: United Kingdom | Date

Fig. 3. Scanmedicine search results page

Users have the possibility to set filters on the side bar and navigate to the visualisation page to view the corresponding graphs as illustrated in Fig 4.

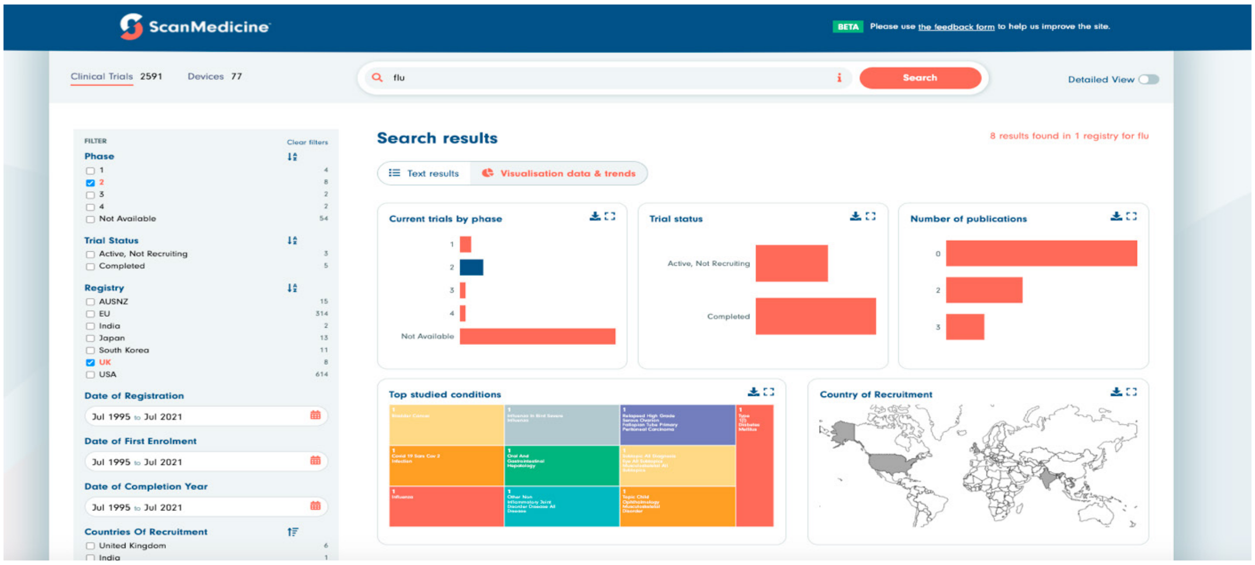


Fig. 4. Scanmedicine visualisation data page

4. Related Work

Several platforms have been launched over the past years to provide accessible resources of clinical trial data information for patients and researchers. Two of these platforms have also included other data sources related to clinical trials e.g BioSamples, PubMed and regulatory agencies however these systems are not operating. Other platforms have restricted their access to few clinical trials registries. In comparison, ScanMedicine is presenting data about clinical trials and medical devices with interactive visualisation.

Clinical Trials Information Mediator (CTIM) is a searching tool developed by Krauth et al [15] to enable users searching linked data in ClinicalTrials.gov, BioSamples and PubMed. The authors introduced the tool as an interface that integrate different resources to allow researchers to identify specific clinical trials that fit one or several search criteria. CTIM was implemented using Spring framework, JavaScript, JQuery and AngularJs. Indexing and searching were developed based on Apache Solr and their system was running in a Linux based server. The system is currently not running, and the interface is not available for use

InfoRoute is another search interface presented by Merbati et al in both English and French to provide access to trails in USA and EU clinical trials registers [16]. The interface manipulates users input by performing query translation and query expansion using UMLS and Health Multiple Terminologies and Ontologies Portal (HeTOP) to generate a new search query. However, the user has to navigate through ClinicalTrials.gov and EU Clinical trials Register websites to explore the results.

Open Trial [17] is another effort made around clinical data sharing platform. Authors aimed to provide access to disparate types and sources of trial information guided by the concept of threaded publications, where all publications related to a trial could be matched together [18]. They consolidated data and documents related to a trial from different resources including academic journals, trials registers and regulatory agencies. Publicly accessible structured data was directly imported into the database and web crawlers were used to access documents that not available for download. They present the matched dataset in two different views: patient and researchers. However, the platform doesn’t display results about emerging trials as the system stopped ingesting new information since 2017.

There is a more recent evolving project provided by the Multi-Regional Clinical Trials Unit (MRCT) to create a single portal through which resources from multiple data-sharing systems and communities can be shared [19]. Their platform hosts a generic trial data repository for both commercial and academic users where investigators, funder, and data generator share patient-level data or meta data to an independent custodian who manages privacy, scientific review and other aspects. The data requester will be bound by the data-use agreements (DUAs) before receiving data.

5. Conclusion

The microservices and serverless paradigms are emerging trend and becoming more popular in the software development industry. They clearly have many advantages over previous architectural approaches, however, there are various elements to be considered when instituting them in the software architecture [20]. Serverless functions minimize operational costs as well as the application deployment cycle and are suitable for stateless, short lived process, whereas microservice are for stateful, long running process. Although serverless computing offers cost reduction in building and running systems, there are cases when it is less expensive to use dedicated servers that are either self-managed or offered as a service. For instance, large applications with a fairly constant, predictable workload [21]. In this paper, we explored the use of both architectures in instituting our solution for designing and implementing a medical searching system. Our proposed architecture highlights the advantages of using microservices together with serverless functions to overcome the current limitations imposed by serverless providers. From our best knowledge, this is the first searching system available to access clinical trials and medical devices in one place, providing corresponding visualisation and analysis features. ScanMedicine is in its first phase, and it is poised for considerable changes over the coming phases for example, extracting data from new documents to build new models, handling the increasing data volumes in all datasets and inferring new knowledge from the integrated data sources by incorporating big data analytics and text mining services. All these emerging requirements drive additional demands around how data is collected, stored, and manipulated. Also, with new developers joining the team in the following stages, a new stack of technologies is expected to be utilized. The architectural design of our system is modular and extensible, and its implementation is compatible with existing commercial and open-source technologies. Such design allows the system to be constantly improved in a flexible and scalable manner.

Acknowledgements

This project is funded by the National Institute for Health Research (NIHR) [HSRIC-2016-10009/Innovation Observatory].

References

[1] World Health Organisation. (2021); https://www.who.int/health-topics/clinical-trials/#tab=tab_1. Accessed 20 April 2021

[2] Krleza-Jeric, Karmela and Trudo Lemmens. (2009) "7th Revision of the declaration of Helsinki: Good news for the transparency of clinical trials." *Croat Med J*. **50**(2): 105-110

[3] U.S. National Institutes of Health. (2021); https://clinicaltrials.gov/ct2/home. Accessed 10 April 2021

[4] Isojarvi, Jaana, Hannah Wood, Carol Lefebvre and Julie Glanville. (2018) "Challenges of identifying unpublished data from clinical trials: Getting the best out of clinical trials registers and other novel sources." *Res Synth Methods*. **9 (4)**: 561-578.

[5] N.J, Halfpenny, Thompson J.C, Quigley J.M and Scott D.A. (2015)"Clinical trials registries for systematic reviews – An alternative source for unpublished data" Value Health 18(3):A12

[6] Baudard, Marie, Amelie Yavchitz, Philippe Ravaud, Elodie Perrodeau and Isabelle Boutron. (2017) "Impact of searching clinical trial registries in systematic reviews of pharmaceutical treatments: methodological systematic review and reanalysis of meta-analyses" *BMJ* 256:j448

[7] Nachtnebel, Anna, Sabine Geiger-Gritsch, Katharina Hintringer and Claudia Wild (2012) "Scanning the horizon – Development and implementation of an early awareness system for anticancer drugs in Austria" *Health Policy* 104: 1-11

[8] World Health Organisation. WHO International Clinical Trials Registry Platform (ICTRP). 2021; https://apps.who.int/trialsearch/. Accessed 20 April 2021.

https://apps.who.int/trialsearch/

[9] Laskey, Kathryn B and Kenneth Laskey. "Service oriented architecture." (2009) *WIREs computational statistics*. **1 (1):** 101-105.

[10] Wolff, Eberhard. "Microservices: flexible software architecture." (2016) *Addison-Wesley Professional*

[11] Adzic, Gojko and Robert Chatley. (2017) "Serverless computing: Economic and architectural impact." *Proceedings of the 11th Joint Meeting on Foundations of Software Engineering* : 884-889

[12] AWS Lambda. 2021; https://aws.amazon.com/lambda/. Accessed 10 April 2021

[13] Cloud Functions, 2021; https://cloud.google.com/functions/. Accessed 10 April 2021

[14] Azure Functions, 2021; https://azure.microsoft.com/en-gb/services/functions. Accessed 10 April 2021

[15] Karakoyun, Christian, Wolfgang Kuchinke, Martin Eckert, Rene Bergmann, Benjamin Braasch, Toresin Karakoyun and Christian Ohmann. (2016) "Clinical trial information mediator." *Journal of Biomedical Informatics*. **63**: 157-168

[16] Merbati, Tayeb, Romain Lelong and Stefan Darmoni. (2015) "InfoRoute: the CISMeF context-specific search algorithm." *Stud health Technol Inform* 216: 544-548

[17] Goldacre, Ben and Jonathan Gray. (2016) "OpenTrials: towards a collaborative open database of all available information on all clinical trials." *Trials*. **17:** 164

[18] Altman, Douglas G, Curt D Furberg, Jeremy M Grimshaw and Daniel R Shanahan. (2014) "Linked publications from a single trial: a thread of evidence." *Trials* **15**:369

[19] Shenoy, Premnath. (2016) "Multi-regional clinical trials and global drug development." *Perspect Clin Res*, **7(2):** 62-67.

[20] Jambunathan, Baskaran and Kalpana Yoganathan. (2018) "Architecture Decision on using microservices or serverless functions with containers." *International Conference on Current Trends towards Converging Technologies* (ICCTCT): 1-7

[21] Eivy, Adam and Joe Weinman. (2017) "Be wary of the economics of serverless cloud computing." *IEEE Cloud Computing*, **4 (2)**:6-12.