ASHIQUR RAHMAN HABEEB RAHUMAN
STUDENT NO: 23352045

Exercise 4 : Implementing a Globally Accessible Distributed Service for WADA

1. REQUIREMENTS

The goal of this exercise is to design and implement a globally accessible distributed system for tracking the whereabouts of the athletes with the World Anti-Doping Agency (WADA). The system would provide a securable scale of platform for the athletes to submit their location information for the Anti-Doping Organizations (ADO's) to access and verify the data shared efficiently.The requirements addressed further ensures that the system not only meets any immediate operational demands but also commits to standard scale of performance,scalability, availability and security.

The distributed service should meet the following requirements:

- **PERFORMANCE**
  The performance of the system should provide a hassle free user experience for both Anti-Doping Agency and athletes with a timely processing of data shared or fetched. To attain the factors mentioned above, inclusion of the metrics such as Response time,Tail latency and System Throughput should be listed specifically.

    1. **Response Time:** The response time should be no more than 2 seconds under normal conditions to handle requests for location submissions and queries.
    2. **Latency:** The latency must be 99 percentile of less than 5 seconds to support operations such as querying athlete location during peak time of slots, ensuring efficient performance even under a high load.
    3. **System throughput**: The system must support a minimum throughput of 1,000 requests per second to accommodate peak times.

- **SCALABILITY**

According to the requirements and the global nature of the WADA's operations, the system must accommodate increasing load.

1. **Horizontal Scaling:** By adding more nodes to the service pool without a downtime in the performance of the system. The system should consistently achieve response times of less than one second for stateless queries, even when operating under heavy load with up to 10,000 concurrent users. Kubernetes clusters will automatically increase their

ASHIQUR RAHMAN HABEEB RAHUMAN
STUDENT NO: 23352045

capacity by doubling the number of nodes when CPU consumption consistently exceeds 70%.

Utilisation of AWS Lambda for rapid scaling of stateless processes and kubernetes horizontal Pod Autoscaler for stateful services based on the memory usage thresholds.

2. **Vertical Scaling:** The performance in the database should support up to 1,000 transactions per second with the read and write latencies not exceeding 10 milli secs during peak load times.

   To ensure the performance scales to be linear with the increased loads, amazon RDS's scaling feature will increase to compute resources during any expected high-load of events.

3. **Dynamic resource Allocation:** The system resources,CPU and Memory , should automatically adjust within a 5 minutes of a spike in demand to aim to keep the system load not more than 80 %.

   Implementation of AWS auto Scaling and cloud monitoring tools such as AWS cloudWatch would dynamically allocate resources based on the real-time system load and performance metrics.

- **AVAILABILITY**

The system should aim for about a 99% uptime,which allows for no more than 4-5 hours of downtime per year. The use of Amazon RDS and EC2 for an auto recovery process will minimise the downtime and ensure a high availability across different geographic locations. Followed by, to implement the redundancy strategy,the data in real-time would be replicated in real.time across at least three geographical regions. The failover instance would be placed in each region that would take over within 60 seconds of detecting a primary instance failure ensuring a continuous operation and data availability, minimising potential service interruptions.

- **RELIABILITY**

Prioritizing the reliability of the service to effectively support the conducted regulations of the Anti-Doping and the first to deal with the integrity of the data, ACID-compliant transactions within AmazonRDS are to make sure that all changes in data maintenance are done with a high standard of accuracy followed by consistency. I find this would be the approach that would support the systems' data integrity, as it would be left unpaired. Further, assuring the transaction processes for a reliability factor, one can use AWS Lambda integrated with

ASHIQUR RAHMAN HABEEB RAHUMAN
STUDENT NO: 23352045

DynamoDB to assure a favorable case. This setup idea guarantees that the operations will be completed fully or in full roll-back; thus, it would avoid any scenario of any partial updates, which would lead to data inconsistencies.It has a service mesh in Kubernetes to effectively orchestrate transactions across all the numerous microservices, and thus the system provides transaction assurance in even a distributed architecture of an environment.The system is self-identifying to the errors and also self-rectifying. It may use AWS CloudWatch, which is another unified logging tool, or even ELK Stack Centralized Logging Technologies to be able to identify and resolve issues on time. The idea is to build in recovery automatically, such as retrying with exponential backoff, so as to handle temporary problems effectively and keep the system functioning unabated. With such personalized notifications, the operations team would be promptly alerted about any perceived reliability risk and, thus, allow them to intervene swiftly to ensure continuous reliability of the system.

This proposal presents an extensive approach for designing a dependable tracking system for WADA, utilising current cloud technology and resilient architecture methodologies.

2. SPECIFICATIONS

2.1. Use Cases

1. **Athlete Registration**: Each of the athletes need to register with the system to start submitting their location data. Essential information of the athletes such as Personal details such ah name, DOB, Nationality, contact information, both email and number, Sports and association details supporting with an authentication medium of setting up a username and password accompanied with an security question for account recovery scenarios for additional verification terms.
2. **Location Submission:** Athletes are required to submit their location once registered at specific intervals to comply with the WADA regulations.
3. **Location Access**: The ADO's must be abl to access the location data of the each athletes who have submitted their location for compliance checks. This would be efficient and secure to support timely anti-doping activities.
4. **Reporting and Analytics:** To assist ADOs in monitoring compliance and spotting trends that can point to doping, the system should offer extensive analytics and reporting features.
5. **Profile Management:** Athletes must be able to update their profile information to keep the data in an updated case.
6. **System Administration:** The superuser's would need to manage the system operations, including user management ,system monitoring, and updates.

2.2. APIs

To implement these use cases, the distributed application exposes the following APIs:

ASHIQUR RAHMAN HABEEB RAHUMAN
STUDENT NO: 23352045

- Register Athlete API
  - **Endpoint:** POST /api/athletes/register
  - **Function**:This API enables the enrollment of new athletes within the system. The main purpose of it is to gather crucial personal and contact details and establish authentication credentials.
  - **Inputs**: Name, Date of Birth, Nationality, Email, Phone Number, Sport, Team, Username, Password, Security Question, and Answer.
  - **Outputs:** Athlete ID, Confirmation Message.
- Location Submission API
  - **Endpoint:** POST /api/locations/submit
  - **Function:** Athletes use this API to submit their current location data as per the compliance requirements of WADA.
  - **Inputs:** Athlete ID, Timestamp, Latitude, Longitude.
  - **Outputs:** Confirmation of data receipt, Message.
- Location Query API
  - **Endpoint:** GET /api/locations/query
  - **Function:** Enables ADOs to access location data for compliance checks. This API supports queries by athlete ID and date range.
  - **Inputs:** ADO ID, Athlete ID, Start Date, End Date.
  - **Outputs:** List of Location Data (Date, Time, Latitude, Longitude).
- Analytics and Reporting API
  - **Endpoint:** GET /api/reports/generate
  - **Function:** Provides ADOs with analytics and reports to monitor athlete compliance and detect potential doping patterns.
  - **Inputs:** Report Type, Date Range, Filters (e.g., sport, region).
  - **Outputs:** Generated Report File (PDF or CSV format).

- Profile Management API
  - **Endpoint:** PUT /api/athletes/update
  - **Function:** Allows athletes to modify their profile information as a way to ensure that the system holds the most reliable and current information.
  - **Inputs:** Athlete ID, Updated Fields (e.g., Email, Phone Number).
  - **Outputs:** Success or Error Message.
- System Administration API
  - **Endpoint:** POST /api/admin/operations
  - **Function:** Enables the operation of various system administrative functions, including user administration, system monitoring, and update implementation.
  - **Inputs:** Operation Type, Parameters.
  - **Outputs:** Operation Status, Messages.

ASHIQUR RAHMAN HABEEB RAHUMAN
STUDENT NO: 23352045

2.3 Actors Interaction

- **Athlete:**
    1. **Registration and Login:** For an athlete, the first process on the platform will be registration. It will require filling out an account with personal and professional details over a secured and intuitive interface. After successful registration, one shall login using the details set during the registration process. The system supports single sign-on capability to allow unbridled access between devices or different sessions.
    2. **Athlete Location Submission:** The main functionality of the athletes is the submission of their whereabouts. This is one of the very important functionalities, and therefore it should be very easy to perform. The submission of the location details can take place by manual entering or GPS integration within their mobile devices. The interface is user-friendly in that it offers clear instructions and real-time feedback to verify whether the data submission was successfully sent, informing the athletes about their compliance instantly.
    3. **Profile Management:** It makes easy to update the athlete's profile and compliance-related information. In this system, it includes various form-based interfaces that come with auto-fill functions for ease and save athlete's time on regular updates.

- **ADO Staff:**
    1. **Dashboard Access:** Accessing the dashboard will mean logging in to a fully interactive graphical interface that presents high-level information with regard to the athlete's compliance data, non-compliance alerts, and various deadlines. This will therefore be highly flexible in the sense that it is tailor-made to meet the demands of each individual staffer and the role they play in the organization.
    2. **Compliance Monitoring and Queries:** Detailed searching of Athlete Location Submissions will be made possible using a dedicated interface. This search interface will support complex queries, be powerful in nature, and at the same time simple to use, supported with dropdowns, date range pickers, and quick filters.
    3. **Reporting and Analytics:** Comprehensive analytical tools allow staff to derive deep insights into the trends and patterns of compliance data. The system gives the possibility for the staff to produce preconfigured reports and prepares one used for their purpose. Such functionality is not only optional but indispensable during strategic planning and operational adjustments within ADOs.
    4. **System Administration**: Admin Users will be able to carry out system settings management and user permissions, which include system updates. All these can

ASHIQUR RAHMAN HABEEB RAHUMAN
STUDENT NO: 23352045

be easily done through powerful back-end management interfaces designed to make the admin more efficient.

- **General Usability:**
    1. Both athlete and ADO interfaces adhere to accessibility standards to ensure that users with disabilities can navigate and use the system effectively.
    2. Recognizing the global nature of the user base, the system is available in multiple languages, which can be easily switched according to the user's preference directly from the interface.

2.3. Failure Model

2.3.1 EXPECTED USAGE AND FAILURE SCENARIOS

The WADA system facilitates a distributed user base, with athletes and Anti-Doping Organizations (ADOs) accessing the platform from diverse places worldwide. The prevalent accessibility of network access brings about various issues, including changes in network latency and vulnerability to outages in regional data centers. To tackle such issues, the system is constructed with a distributed architecture that leverages cloud services to enhance the transfer of information and provide seamless operation. Athletes regularly submit their location data using the Location Submission API (POST /api/locations/submit), requiring the system to effectively handle a large number of write operations. Similarly, ADOs often use the Location Query API (GET /api/locations/query) to get this data for compliance evaluations, which requires substantial backing for concurrent read operations. To maintain optimal performance in such situations, AWS Auto Scaling is utilized to dynamically adapt resources based on the workload, whereas Amazon RDS read replicas are employed for spreading the query load. Moreover, the regular usage of athlete profiles for changes via the Profile Management API (PUT /api/athletes/update) requires efficient caching techniques to minimize the load on the database. The design accounts for uncommon, although potential, faults such as server outages or software glitches, which are addressed by extensive monitoring using AWS CloudWatch and the setup of Kubernetes to manage the resilience of microservices.

2.3.1. Failures handled by the application

Particular steps are taken to control faults in user interaction and data integrity at the application level. The procedure for registering using the Register Athlete API (POST /api/athletes/register) incorporates comprehensive input validations to prevent the submission of wrong information that could end up in further problems. The system leverages AWS Lambda to execute stateless components seamlessly and employs Amazon RDS for data transactions to maintain ACID compliance, improving consistency and dependability throughout operations.

ASHIQUR RAHMAN HABEEB RAHUMAN
STUDENT NO: 23352045

2.3.2. Failures handled by the managed cloud services

1. **Handling Network Disruptions and Server Failures:**

   The Amazon EC2 Auto-Recovery is indispensable in managing server failures. Amazon EC2 has persistent monitoring to allow the detection of the problematic instances that may have existed in the failure-to-respond-through-network condition. For these types of events, EC2 will initiate recovery processes by itself, which will include either instance reboot or migration to a healthy host. This quick response to the prompt helps to minimize the duration of time the system is not functioning and assures uninterrupted service without manual intervention.

   The Amazon RDS works in geographical sites as a control for data center failure risk for data recovery. This multi-region deployment entails the replication of databases between more than one AWS region in such a manner that the data is redundant, and at the same time, the retrieval speed for various user locations is optimized. For instance, in the case of failure within one area, failover to the database replica within the other area can be made in order for the application to remain operational and for the data to stay available. WADA athlete tracking system runs on a global infrastructure of Amazon Route 53.

2. **DNS Routing and Content Delivery with Amazon Route 53 and AWS CloudFront:**

   It is charged with the mandate of ensuring an effective and efficient management of DNS and enhancing user access. DNS web service Route 53 is effective in making the translation of human-friendly domain names to the required network connection IP addresses. Geolocation routing by Route 53 is important for athletes and an ADO-based system that is being used all over the world. This simply means that the request for data usually gets forwarded automatically to the nearest server with relation to the geographic location of the user. For example, in a scenario where an athlete has uploaded data, this functionality would help either through the POST request of Location Submission API **(POST /api/locations/submit**) or through the ADO needing a location retrieval feature of its Location Query API **(GET /api/locations/query)**. Route 53 decreases the reaction times and system efficiency by reducing the latency and potential bottlenecks through the traffic optimization of routes for the network.

2.4. SECURITY AND DATA PRIVACY COMPLIANCES WITH SPORTS INDUSTRY

The architecture assumes secured API protocols that guarantee, in a mutually exclusive manner, using connections conducted over HTTPS. It assumes data at rest encryption and the same

ASHIQUR RAHMAN HABEEB RAHUMAN
STUDENT NO: 23352045

when it is in transit across the network. This encryption is enabled through AWS services like RDS and S3, as per the guidelines of the System Administration API. I outlined a way to integrate AWS Cognito to manage user authentication and access in the right way. This would allow strong access control through registration of the athlete API and session administration. When supplemented by regular security checks and compliance surveillance, this solution would be adjusting to threat changes and would allow adherence to regulations such as GDPR. This further ensures that substantial actions such as data reduction and purpose limitation are in place; thus, all the above play a very crucial role in operations as stated in the Analytics and Reporting API and the Profile Management API. The design will also provide secure and compliant data transfer, crossing boundaries in such a way that assures legal normalization of all transfers of international data, ensuring reliable and secure systems in its operations around the world. It will be strong and secure, guiding during any future operational or deployment issues with a comprehensive approach taken during the conceptual phase. The Location Query API ensures integrity and access control.

Based on the architecture of the system set for the World Anti-Doping Agency, it meets very severe laws in data protection, which includes GDPR, along with the regulation prevailing in the sports industry. What matters most is that all the transmitted and stored information for the athletes takes place through the help of Amazon Web Services, thus very secure. The architecture is GDPR-compliant, through explicit user-consent mechanisms at the time of registration and simplified procedures of access, rectification, and erasure of data, so that an athlete can manage his personal information easily. Even for an athlete, he or she can have the automated feature in the watch, and that means portability and receiving data in a standardized format is made easier.

ASHIQUR RAHMAN HABEEB RAHUMAN
STUDENT NO: 23352045

3. Architecture

3.1. Technical Architecture

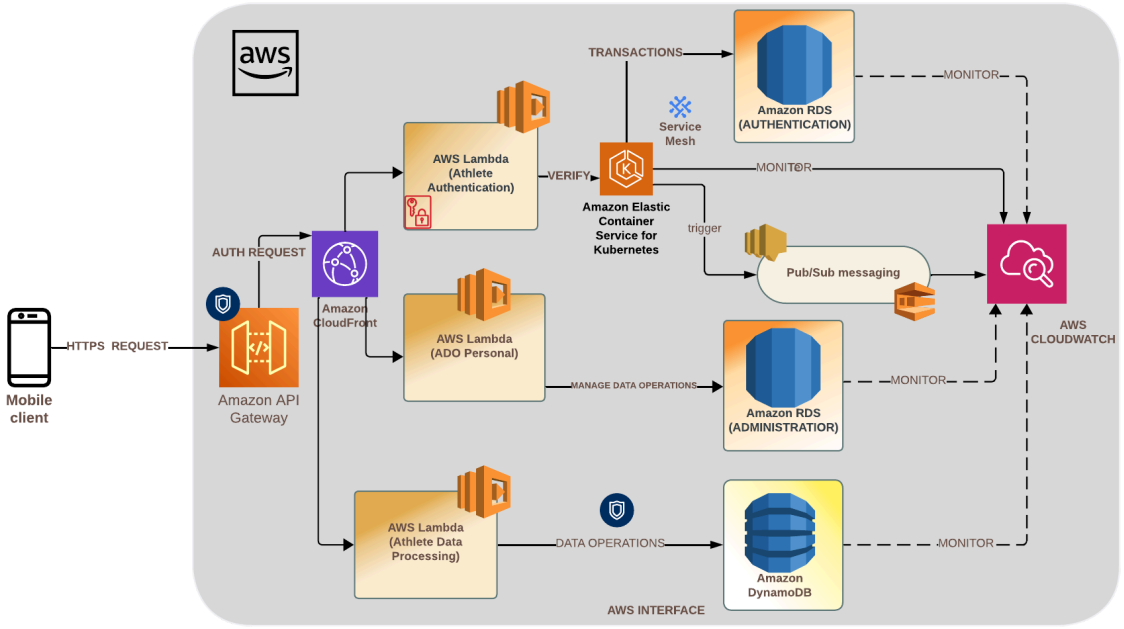

Figure 1: Technical architecture diagram

3.1.2 Addressing Functionalities

Designed architecture adeptly addresses the performance, scalability, availability, and reliability in a manner that will support the globally distributed system of the World Anti-Doping Agency. This will further enhance performance in that, together with Amazon API Gateway, AWS Lambda will be used. This will hasten in-demand responsiveness to user requests at the quickest possible time, hence achieving maximum data throughput. So, it optimizes the system response time to be within two seconds in normal conditions and still enables effective handling of critical operations, easing location updates and location queries, even under peak loads.

The pure stateless AWS Lambda functions and the stateless Amazon Elastic Kubernetes Service (EKS) further assure the ability to scale and opens the processing ability for horizontal scaling across multiple instances based on demand. Using Amazon DynamoDB in the storage of non-relational data will yield an advantage due to high performance while retrieving and storing data, which will auto-scale with the volume of data or the number of requests. This will support 10,000 concurrent users without performance degradation.

ASHIQUR RAHMAN HABEEB RAHUMAN
STUDENT NO: 23352045

This is in line with deploying the system to high availability through multiple AWS Availability Zones. Amazon RDS and DynamoDB have been set up with multi-region replication and failover strategies. This enables the possibility of data reachability even in case of disruption of regional AWS service. This kind of setup ensures operating uptime of 99% in adherence to strict standards for availability.

Amazon RDS ensures that the data integrity and transaction reliability of the users is supported, including complex queries with transactional data that allows ACID compliance. Further enhanced reliability is added through AWS managed services, which are very robust and stable. This enhances further the resilience of the system through tracking the performance of the systems in real-time and fixing any likely problems proactively using detailed monitoring and logging with AWS CloudWatch.

3.1.3 Interactions between Actors

- **Athlete Login and Location Submission:**
  - **Athlete Login:** When an athlete wants to access the system to submit his location information, he will initiate a login request from his mobile client. The request is securely sent to Amazon API Gateway, which routes the request through to Amazon Cognito for authentication. Cognito is going to validate the athlete's credentials, and if everything is successful, it will return a session token back to the athlete's device. This token will be necessary to secure subsequent interactions with the system.
  - **Location Submission:** After the athlete is authenticated, the athlete goes ahead to submit his location data. The submitted location data, together with the session token, is then sent back to the API Gateway. In response, the gateway will check the token, this time against Cognito, to ensure the session is still active and secure. After successful validation, the request passes through an AWS Lambda function specific to handling athlete data. The function processes the athlete location data and writes it to Amazon DynamoDB with proper time stamps and an athlete ID. Data is efficiently stored; it has a very fast time to write, and it is available for querying immediately.

- **ADO Staff Login and Data Query:**
  - **ADO Staff Login:** Anti-Doping Organization (ADO) staff also log in via a similar process as athletes. They use their credentials to access a dedicated interface for compliance monitoring. After successfully authenticating with Amazon Cognito, the members of staff are allowed into the system in order to be able to carry out checks on compliance and queries on data.

ASHIQUR RAHMAN HABEEB RAHUMAN
STUDENT NO: 23352045

- **Query Handling:** ADO staff would inquire the location-registered data on the athletes for compliance monitoring of doping. In case the query is done, it travels via the API Gateway, routing the request to another lambda function designed to handle complex queries. This part of the function retrieves all the needed data from Amazon RDS and uses its strong transactional capabilities in maintaining data integrity and accuracy. This then could provide the system with the possibility to process more complicated queries of the type that will involve different parameters—date ranges and, say, athlete ID numbers or other numbers pointing to certain events.
- **Performance and Scalability:** Now, During major sports events, which one can expect location submissions and queries pressure to surge up. So, both AWS Lambda and Amazon EKS will scale elastically with the variation in pressure. Amazon CloudFront: it is the fastest and globally used content delivery network service that offloads the delivery of data from origin servers to many cache servers distributed in several world locations.

CONCLUSION

The above-proposed architecture of this globally accessible distributed system can indeed meet the WADA requirement rigorously and keep track of athlete whereabouts with high performance, scalability, availability, and reliability. Its architecture uses advanced AWS services that enable this system to bear large loads without loss and support data integrity in the distributed network. The strict security of data includes the international conformance of all GDPR standards. More enhancements will add features that make the system user-friendly and exciting for use. The system stands out as one of the robust, scalable, and secure solutions prepared to play a critical contribution toward maintaining integrity in sports for a doping-free environment.

ASHIQUR RAHMAN HABEEB RAHUMAN
STUDENT NO: 23352045

REFERENCES

(1) https://docs.aws.amazon.com/whitepapers/latest/microservices-on-aws/microservices-on-aws.html

(2) https://www.splunk.com/en_us/blog/learn/distributed-systems.html

(3) https://www.wada-ama.org/en/athletes-support-personnel/provide-whereabouts,

(4) Takabi, H., Joshi, J.B.D., & Ahn, G.-J. (2010). Security and privacy challenges in cloud computing environments. IEEE Security & Privacy, 8(6), 24-31.

ASHIQUR RAHMAN HABEEB RAHUMAN
STUDENT NO: 23352045

DECLARATION

I understand that this is an individual assessment and that collaboration is not permitted. I have not received any assistance with my work for this assessment. Where I have used the published work of others, I have indicated this with appropriate citation.

I have not and will not share any part of my work on this assessment, directly or indirectly, with any other student.

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at http://www.tcd.ie/calendar.

I have also completed the Online Tutorial on avoiding plagiarism 'Ready Steady Write', located at http://tcd-ie.libguides.com/plagiarism/ready-steady-write.

I understand that by returning this declaration with my work, I am agreeing with the above statement.

Name: ASHIQUR RAHMAN HABEEB RAHUMAN

Date: 31/04/2024