



Modular Community Management System (MCMS)

Adam Cornfield, st20226675

Advanced Programming SEN6000

13/11/2025

Case Study Proposal

What is the problem currently?

- **Communities typically have old disjointed systems and tools**
 - Spreadsheets, outdated websites etc
- **Poor Compliance with GDPR and DPA**
 - Many communities aren't even aware of many details within GDPR
- **Limited budget and technical skill**
 - Can't pay for expensive CRM solutions or bespoke developer
- **Existing solutions force conformance with set standards**
 - Groups with specific requirement can't adapt easily
 - Locks in communities to specific providers, the "explicit lack of standards to support portability and interoperability among cloud providers stifles the market competition and locks customers to a single cloud provider" (Opara-Martins et al., 2016)



Case Study Proposal

What can MCMS do to fix this?



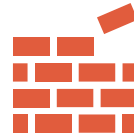
Open Source model

Allows **community to contribute** reducing **development cost**

Users have **direct say** in development and can suggest or provide their own changes to the core code

Not dependant on single developer or organisation for development and **keeping system up to date**

Avoids issues surrounding vendor lock-in



Modular Design

Allows for **new features to be added** quickly and easily

System can be customised to **exact specification** of a group



Designed for modern standards

Cybersecurity Best Practice (SHA-256 Encryption, MFA, Authentication Providers)

WCAG 2.2 AA compliant

Built in **localisation support**

Compliant with **GDPR** and **Data Protection Act**



Community Owned

Communities own the infrastructure and the code

Not dependant on third party providers to maintain the code and service

Primary Users

Community Managers

- Responsible for organisation and day to day operations of a group

Events Coordinators

- Create and manage events for the group

Individual Members

- Primary consumers of content and main source of interaction

Secondary Users

Module & Open Source Developers

- Main contributors to improvement of the platform

Webmasters and Hosting platforms

- Responsible for deployment of MCMS and technical configuration for groups

Prospective Members & Visitors

- MCMS may be their first experience with a group
- Should be inviting and easy to navigate

Stakeholders

Student Unions

- MCMS can be used by individual student societies
- The unions would have an interest in the success these platforms bring to their societies

Councils

- A vibrant community is in the local government's best interest

Umbrella Organisations

- Any organisation that exists to promote community activates

Scope

IN SCOPE

Member onboarding

- Registration of new users
- Information capturing

Event Management

- Calendar feature
- Basic ticketing
- Reminders

Record and File management

- Store meeting minutes, policy documents and so on

Core modularity

- Compatibility with additional modules that extend the core features of the code

Basic list of pre-approved modules

- Both official and un-official modules
- Link to websites such as GitHub and NPM

OUT OF SCOPE

AI Large Language Model Integration

- AI features will not be included
- It would be useful for initial config and task automation
- Very difficult to implement

Module Marketplace

- Scale and complexity of this would be immense and not realistic for this project scale
- Would require complex systems to moderate modules and ensure malicious code is not published

Payment processors

- To get setup with payment processors (such as stripe) requires a business to be set up, this would not be possible at this stage

Hosting and Deployment

- MCMS is a template, not SaaS, MCMS does not come pre-packaged with hosting
- This could become a feature for a business based around MCMS however

SYSTEM COMPONENTS & CORE FUNCTIONALITY (FUNCTIONAL REQUIREMENTS)

1. Essential Functions and User Management



User Registration and Authentication

Full login system with Oauth and MFA for security
Registration form for initial data collection



Administrator Portal

View memberships, member data, modify data, view community statistics
Full auditable logs for viewing, editing and modification of user data



User access controls

Full customisable permissions system
Integration with all modules, such that all permissions are managed centrally

2. Events Management



Event Viewing

Calendar View
List view sorted by date
Full events detail page with location, date, time and other information
Uploading banners for added visual effect



Ticketing

Basic ticket functionality, allowing for users to sign up to events
Future capability to pay for events



Event management

Interface to allow for uploading, editing and deleting of events
Customizable by modules



Waitlists

Allow people to queue for popular events for secondary ticket sales



Reminders

Options for email reminders to be sent to users

3. Document Management



Page to view all documents related to group

Things such as policy documents, meeting minutes, guides or blogs created directly on site.



Document editing

Basic interface for certain users to create content to publish as a blog style



Document Upload

Portal to allow for documents to be seamlessly uploaded and stored on the webserver
Integration with permissions system for access to files

4. Module Integration

Documentation on setup process of MCMS

- This includes guides on how to install and include the modules themselves

System capable of integration with third party modules

- The system should be compatible with third party modules and integrate seamlessly

Security features to control module access to data

- Webmaster must specify for each module, what data and what core systems it is able to access

Implementation Plan

Core Technologies

- **Node.js + Express.js for webserver**
 - Well supported with lots of documentation
- **EJS (Embedded JavaScript) + Bootstrap Front End**
 - Open source tools which are well documented and with limited learning curve
- **MYSQL Database for data storage**
 - Relational database structure and easily adoptable for anyone with SQL skills
- **Authentication**
 - Passport.js for cookie management
 - MFA (Multifactor Authentication)
 - Ensures data security by providing an extra layer of security
 - Password Encryption
 - OAuth2 integration with google, Facebook, etc...
 - Reduces risk by giving responsibility to larger organisations with proven track records

Implementation Plan

Core Technologies Continued...

- **Developed in VS Code**
 - Large marketplace so developers can fine tune their experience
 - Integrates with Git seamlessly
 - Modern and clean interface with many powerful features to assist code development
- **GitHub Version Control**
 - Provides accurate change history for the whole project
 - Allows other developers to easily contribute to source code
 - Repository forking and pull requests
 - Automation of CI/CD pipeline via GitHub actions
- **Nginx Reverse proxy**
 - New, modern and fast
 - Allows full control of the domain
 - Scales well with load balancers to distribute load

Implementation Plan

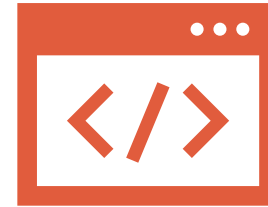
Why these technologies?



Popular languages and frameworks

Well supported with lots of community contributions and bug reports

Lots of high quality packages that can be added to speed up development via NPM



Modular Architecture

Express routing is modular by nature, making it easy to integrate

NGINX is also modular and flexible allowing for hosting different sites on the same hardware

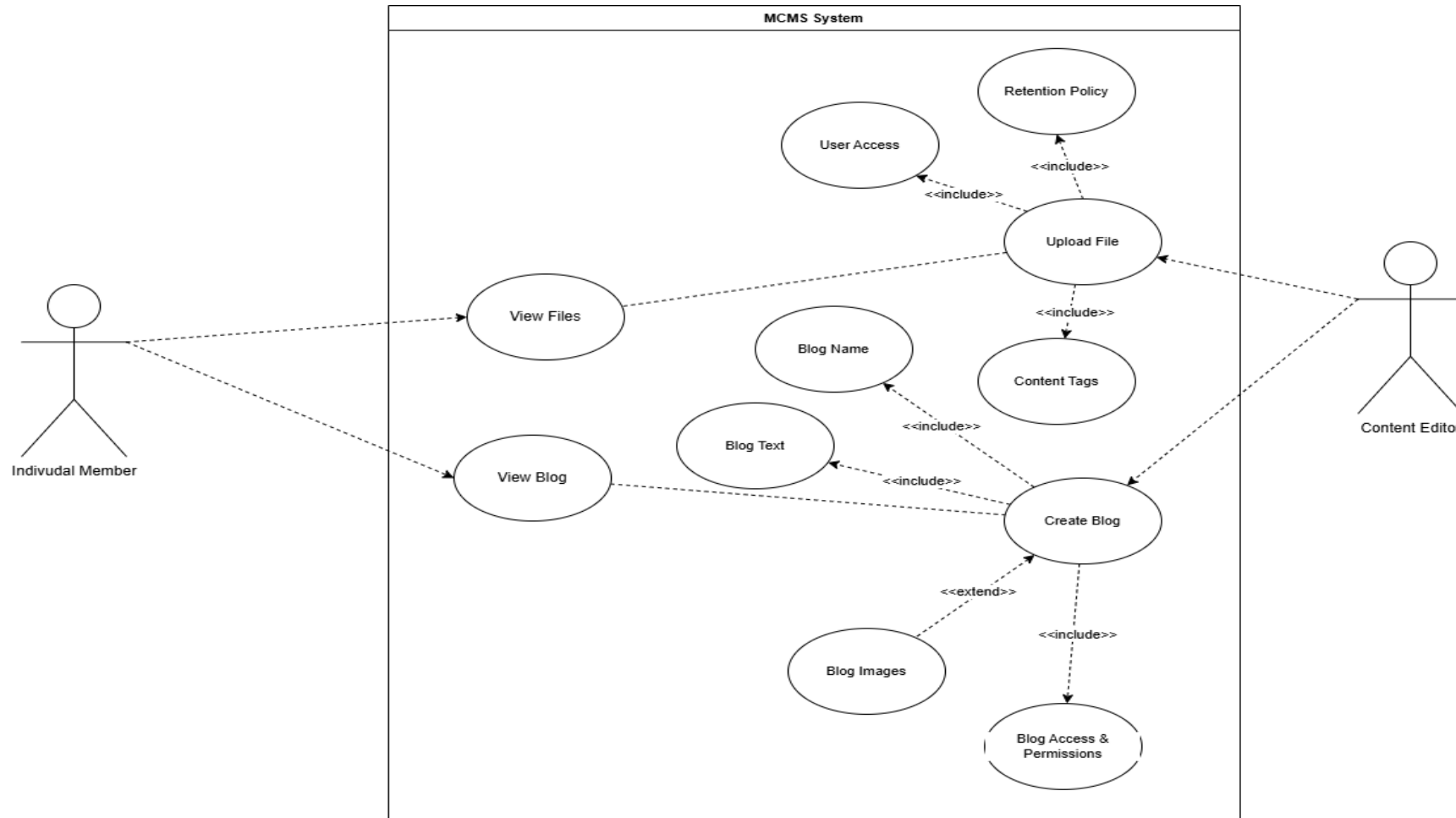
Easy load balancing

INITIAL DESIGN CONCEPT

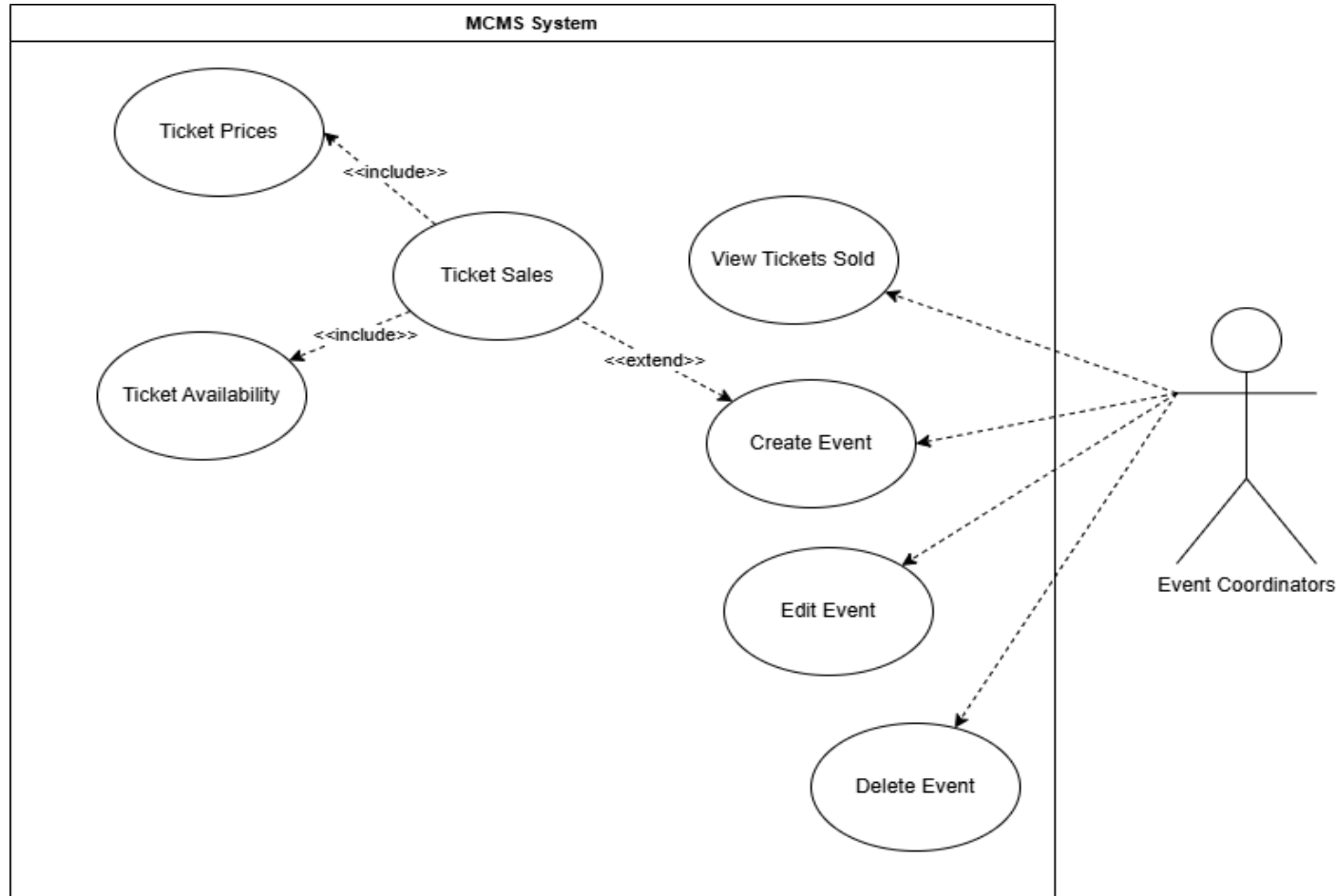
Community Managers



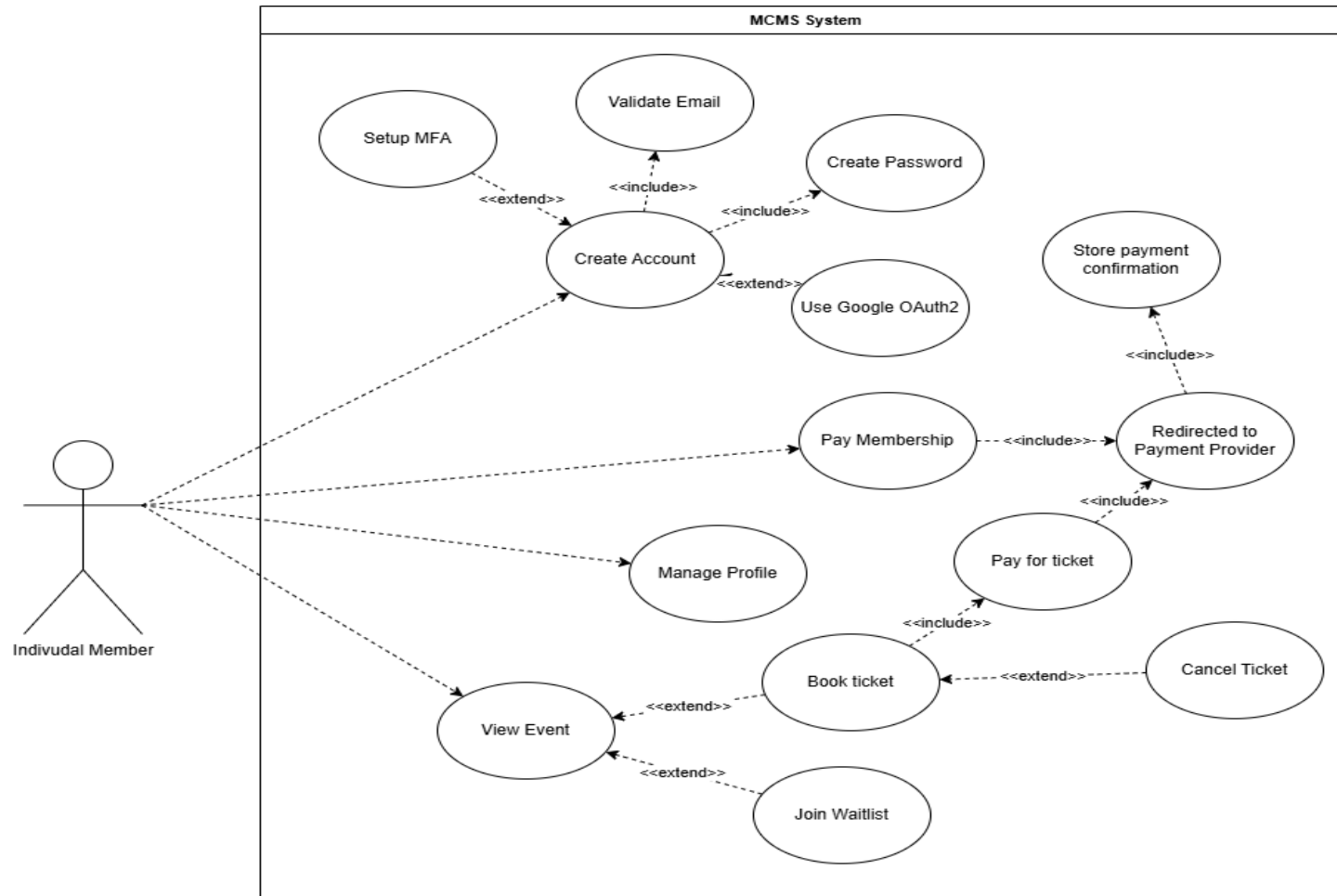
Content Editor Use Case



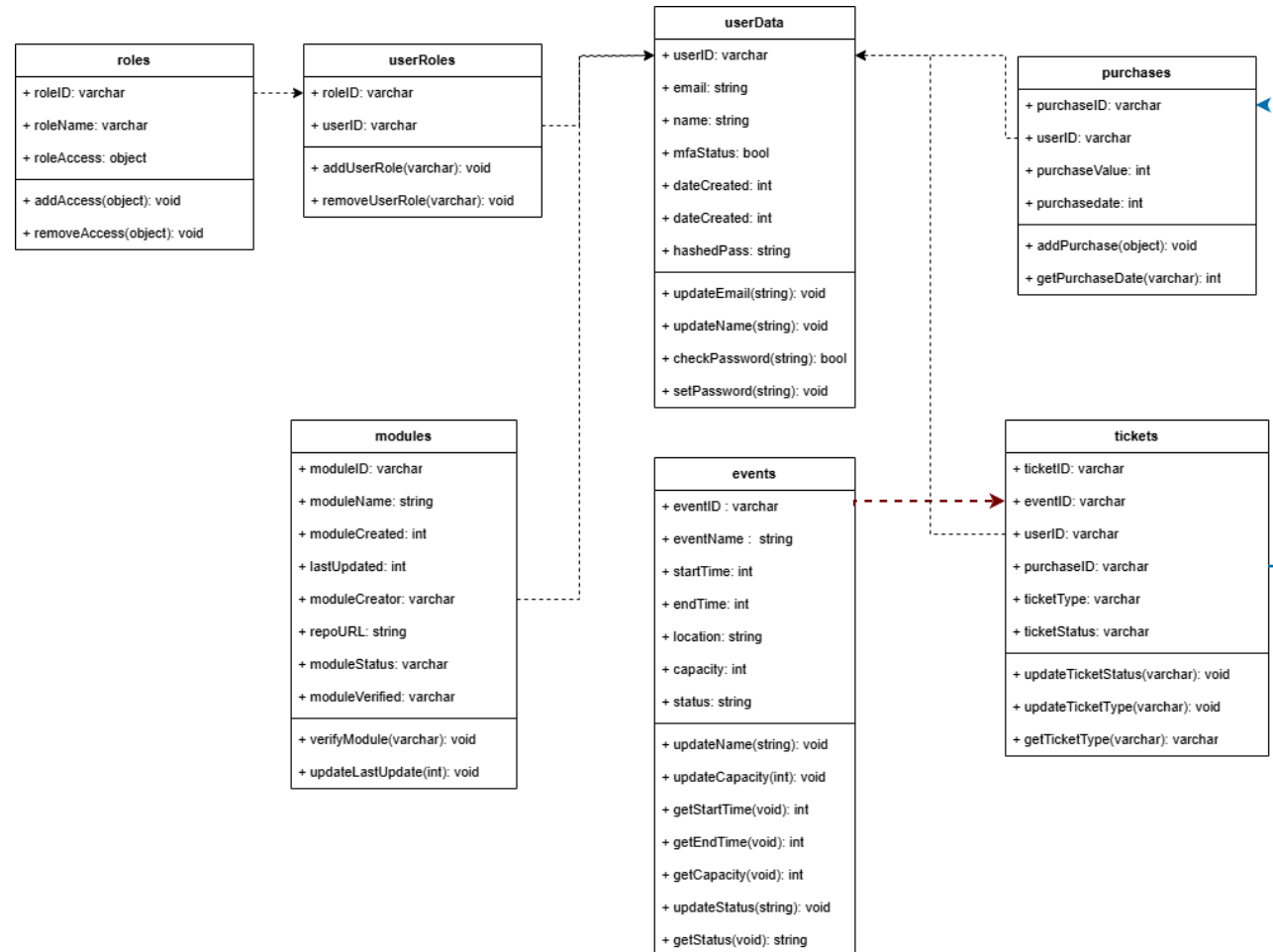
Event Coordinator Use Case



Individual Member Use Case



Class Diagram



Summary



MCMS provides a highly customisable base for all communities



Usable regardless of specific requirements



Prevents provider lock in



Assists in compliance with GDPR, DPA and WCAG Standards

AI Acknowledgement

- I acknowledge the use of ChatGPT-5 in this work (Open AI, <https://chat.openai.com/>), usage is limited solely to helping provide general structure suggestions, feedback and assistance in finding relevant research papers for referencing.
- All references provided via ChatGPT have been separately fact checked that quotes are correct
- All content in this presentation is my own, including written text and images except where otherwise referenced.

References

- *Web content accessibility guidelines (WCAG) 2.2* (2024) W3C. Available at: <https://www.w3.org/TR/2024/REC-WCAG22-20241212/> (Accessed: 12 November 2025).
- Opara-Martins, J., Sahandi, R. and Tian, F. (2016) *Critical analysis of Vendor Lock-in and its impact on cloud computing migration: A Business Perspective - Journal of Cloud Computing, SpringerOpen*. Available at: <https://journalofcloudcomputing.springeropen.com/articles/10.1186/s13677-016-0054-z> (Accessed: 12 November 2025).

QUESTIONS