

Cloud-Native License Management System for Hybrid Cloud Using Azure Arc

Expanded Research & Professional Portfolio Project

Author: Okwuiwe Alphonsus Jonas

Discipline: Cloud Engineering | Cybersecurity | Hybrid Infrastructure | DevOps

Abstract

This study presents the design, development, and evaluation of a cloud-native license management system intended for hybrid cloud infrastructures. Many organizations operate mixed environments consisting of on-premise servers, virtual machines, and public cloud resources, which complicates software asset governance. The proposed solution integrates Azure Arc for unified control, automated software discovery agents, RESTful microservices, and a centralized analytics dashboard. By applying cloud-native design principles such as containerization, scalability, and Infrastructure-as-Code, the platform provides real-time visibility, compliance assurance, and cost optimization. The system demonstrates how modern governance strategies reduce operational risk and financial waste while improving audit readiness (Microsoft, 2024; NIST, 2020).

1. Introduction

Software licensing is a critical yet frequently overlooked component of IT governance. Organizations often deploy applications across multiple servers without maintaining accurate records of consumption. As infrastructure scales, manual tracking using spreadsheets becomes error-prone and inefficient. This leads to license oversubscription, legal penalties during audits, and unnecessary spending on unused assets. With the rise of hybrid cloud computing, these challenges intensify due to distributed control and inconsistent management practices. This project proposes an automated, cloud-native platform capable of collecting, storing, and analyzing software usage metrics from all connected systems. The goal is to deliver a scalable, secure, and cost-effective approach to enterprise license governance (Gartner, 2023).

2. Problem Statement

Traditional software asset management tools are often expensive, complex, or limited to a single environment. Small and medium organizations may lack the budget for commercial platforms, while large enterprises struggle with integration across multiple clouds. The absence of centralized visibility creates risks including compliance violations, shadow IT, and inefficient procurement decisions. Therefore, a lightweight, cloud-native, and extensible system is required to address these operational gaps.

3. Literature Review

Cloud-native computing promotes the use of microservices, APIs, and containers to enhance resilience and scalability (Burns & Beda, 2019). Hybrid governance frameworks recommend unified management planes that provide consistent policies across environments (NIST, 2020). Azure Arc extends cloud control features to on-premise resources, enabling centralized monitoring and policy enforcement (Microsoft, 2024). Research from Flexera (2023) and Gartner (2023) highlights that automated software asset management can significantly reduce license costs and improve compliance rates. These findings support the design decisions adopted in this project.

4. Methodology

The project followed a practical engineering methodology. First, requirements were gathered based on common enterprise challenges. Second, a modular architecture was designed to separate data collection, processing, and presentation layers. Third, open-source technologies were selected to ensure cost-effectiveness and portability. Finally, the system was tested in simulated hybrid environments consisting of local machines and virtual servers. This iterative development approach enabled continuous improvement and validation.

5. System Architecture

The architecture is composed of five primary layers: (1) Hybrid server layer connected via Azure Arc agents, (2) Data collection scripts that discover installed software, (3) Backend API services for processing and storage, (4) Database layer for persistence, and (5) Frontend dashboard for visualization. This layered structure ensures scalability, maintainability, and clear separation of responsibilities. Containers enable consistent deployment across different platforms.

6. Implementation and Technical Design

PowerShell Discovery Agent:

```
$software = Get-ItemProperty HKLM:\Software\Microsoft\Windows\CurrentVersion\Uninstall\* |  
Select DisplayName, DisplayVersion |  
Where-Object { $_.DisplayName } |  
ConvertTo-Json
```

FastAPI Backend Example:

```
from fastapi import FastAPI  
app = FastAPI(title="License API")  
  
@app.get('/health')  
def health():  
    return {"status": "ok"}
```

Docker Configuration:

```
FROM python:3.11  
WORKDIR /app  
COPY . .  
RUN pip install fastapi uvicorn  
CMD ["uvicorn", "main:app", "--host", "0.0.0.0", "--port", "8000"]
```

7. Results and Evaluation

Experimental deployment demonstrated that the system successfully collected software metadata from multiple machines and stored the results centrally. Alerts were generated when license limits were exceeded, allowing proactive remediation. Compared to manual tracking, the automated approach reduced administrative time and improved accuracy. The modular design also allowed quick scaling to additional nodes without performance degradation.

8. Security Considerations

Security was prioritized throughout development. Communications are encrypted via HTTPS, and authentication mechanisms restrict unauthorized access. Role-based access control ensures that only authorized personnel can modify license data. Logs are centralized to detect anomalies and support forensic analysis. These controls align with zero-trust principles and modern cybersecurity best practices (NIST, 2020).

9. Future Enhancements

Future improvements may include predictive analytics for license forecasting, integration with procurement systems, Kubernetes orchestration, automated compliance reports, and machine learning models for usage optimization. Multi-tenant capabilities could transform the solution into a Software-as-a-Service offering for managed service providers.

10. Conclusion

The project demonstrates that a cloud-native, hybrid-compatible architecture can effectively manage software licenses at scale. By combining automation, centralized analytics, and secure governance, organizations can reduce costs and compliance risks. The research highlights the practical benefits of Azure Arc and modern DevOps practices in enterprise IT environments. The system serves as both an academic case study and a professional portfolio artifact showcasing cloud engineering competence.

References

- Burns, B., & Beda, J. (2019). Kubernetes: Up and Running. O'Reilly.
- Flexera. (2023). State of Software Asset Management Report.
- Gartner. (2023). IT Asset Optimization Research.
- Microsoft. (2024). Azure Arc Documentation.
- NIST. (2020). Security and Privacy Controls for Information Systems.