LowTech GMmBH Techincal Transformation Milestone 1

Wladymir Alexander Brborich Herrera
wladymir.brborich-herrera@stud.fra-uas.de,
Vishwaben Pareshbhai Kakadiya
vishwaben.kakadiya@stud.fra-uas.de,
Hellyben Bhaveshkumar Shah (1476905)
hellyben.shah@stud.fra-uas.de,
Heer Rakeshkumar Vankawala (1449039)
heer.vankawala@stud.fra-uas.de, and
S Priyanka Dilipbhai Vadiwala
priyanka.vadiwala@stud.fra-uas.de

Frankfurt University of Applied Sciences
(1971-2014: Fachhochschule Frankfurt am Main)
Nibelungenplatz 1
D-60318 Frankfurt am Main

Abstract In this report, we as consultants from *Awesome Cloud AG* present a technical transformation analysis aimed at modernizing the infrastructure of *LowTech GmbH*, a small to medium-sized enterprise specializing in wooden furniture production. The analysis includes a critical assessment of the current infrastructure, energy consumption calculation for the existing setup followed by a detailed transformation roadmap of future-ready modern infrastructure and explanations of enhancements in scalability, availability, and security compared to current infrastructure. This analysis will serve as a foundational step for subsequent project phases, ensuring that *LowTech GmbH* is well-equipped to meet future requirements and challenges.

This is where the introduction (the prologue or foreword) comes in. The introduction should also be short and concise. The reader should be prepared for the text that follows. Of course, the introduction should also be formulated in an interesting way.

1 Overview of the problem

LowTech GmbH has seven departments with all of their infrastructure hosted on premise. The original partner in charge of the infrastructure has gone out of business

2 Objectives of the technological transformation

The main high level objective is to migrate the current infrastructure into a private-cloud context, meaning that we are going to rent server space and manage only the software components of the technology stack. The hardware maintenance and provisioning will be handled by a third party. We expect to have an improvement in performance just by the fact that we will be using newer/faster technologies, nevertheless, we expect at least parity with the current implementation.

2.1 Scalability, availability and security analysis

3 Assessment of the current (as-is) infrastructure

3.1 Current traffic and usage

3.2 Scalability, Availability and Security Analysis

According to NIST definition of cloud computing is given as "Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." [2]

As *LowTech GmbH* is based on Legacy IT infrastructure and currently lacking *NIST five essential characteristics* such as On-demand self-service, Broad network access,Resource pooling,Rapid elasticity and Measured service.

Elasticity / Scalability

- **Fixed Hardware & Inflexible Infrastructure**:
  Current infrastructure consists of 7 on-premises servers housed in a single 19-inch rack, along with 17 clients and 19 laptops all with predetermined, static configurations. Moreover, the physical constraints of the on-premises setup, with no additional space for expansion, severely limit scaling options. This inflexibility makes it challenging to accommodate growth or adapt to changing business needs.

- **Absence of Resource Utilization/pooling**:
  There's no apparent way to quickly scale resources up or down based on demand or user traffic fluctuation in the current infrastructure. Each application typically runs on a dedicated server with fixed resources. This approach leads to inefficient resource utilization, as some servers may be underutilized while others are overloaded which may lead to performance issues during peak times or resource waste during low-demand periods due to no dynamic resource allocation.

- **Manual Processes & High Cost**:
  Any changes in capacity would likely require manual hardware upgrades or replacements including hardware installation, and configuration, making the process time-consuming and potentially leading to downtime. Replacement of hardware is not only tedious but also financially burdensome due to high costs of new hardware.

Availability

- **Obsolate Hardware/OS & Runtime Environments**:
  Many components of the current infrastructure is based on very old hardware and outdated operating systems such as Windows XP SP3 (Finance clients), Windows 7 SP3 (HR clients and Customer Service laptops), Debian 5.0 Lenny (Warehouse clients and server), Ubuntu 16.04 LTS (Sales CRM Storage server) etc. Several applications are running on outdated software versions such as Java 1.7/1.8, MySQL 5.5/5.7, PHP 5.3 and Firefox 3.6 etc. which makes this whole infrastructure more susceptible to failure.

- **Lack of Redundancy and Backup Mechanism**:
  There's no mention of redundant systems or data backup solutions which might lead to significant service disruptions as well as data loss in case of any system failure.

- **Manual Maintainance**:
  It is impossible to meet high availability requirements without a robust failover mechanism due to manual maintenance operations. This increases the possiblity of human errors, leads to longer downtime and reduces overall reliability.

Security

- **Basic Windows Firewall and pSense**:
  Many components of the current infrastructure is based on very old hardware and outdated operating systems such as Windows XP SP3 (Finance clients), Windows 7 SP3 (HR clients and Customer Service laptops), Debian 5.0 Lenny (Warehouse clients and server), Ubuntu 16.04 LTS (Sales CRM Storage server) etc. Several applications are running on outdated software versions such as Java 1.7/1.8, MySQL 5.5/5.7, PHP 5.3 and Firefox 3.6 etc. which makes this whole infrastructure more susceptible to failure.

- **Lack of Redundancy and Backup Mechanism**:
  There's no mention of redundant systems or data backup solutions which might lead to significant service disruptions as well as data loss in case of any system failure.

- **Manual Maintainance**:
  It is impossible to meet high availability requirements without a robust failover mechanism due to manual maintenance operations. This increases the possiblity of human errors, leads to longer downtime and reduces overall reliability.

3.3 Energy consumption and approximate cost

Energy consumption calculation for the as-in infrastructure of Low Tech GmbH is as follows :

Departments	**Server (Qty x Power)**	**Client (Qty x Power)**	**Laptop (Qty x Power)**	**Total Power Consumption**	**Annual Energy Consumption(KWh)**
Finance	1 x 1000W	4 x 500W	-	3000W	26,280
HR	1 x 1000W	3 x 500W	-	2500W	21,900
Warehouse	1 x 1000W	10 x 500W	-	6000W	52,560
Sales	1 x 1000W 1 x 1200W	-	10 x 50W	2700W	23,652
Operations	1 x 1200W	-	4 x 50W	1400W	12,264
Customer Service	-	-	5 x 100W	500W	4,380
Webshop	1 x 1200W	-	-	1200W	10,512

Table 1: Power Consumption by Department and Device Type

Total Energy Consumption (Annual) : 151,548 KWh (151.548 MWh)

According to Eurostat published data of electricity prices for non-household consumers [1], Low Tech GmbH falls under the annual energy consumption band 'IB (20 MWh to 499 MWh)' with energy price 0.3244 € per KWh.

Total Cost for Energy Consumption (Annual) : 151,548 KWh x 0.3244 € = 49,162.17 €

4 Client Requirements

- Make the infrastructure more scalable, available and secure

- Modernize the technology, runtime and operation modality
- Perform the migration with a maximum downtime of 4 hours
- Cost reduction
- Maintenance reduction
- High availability of 99.5%

5 Assessment of potential technological components

5.1 Hardware

- Provisioning modern hardware locally
- Renting server space

5.2 Virtualization technologies

5.3 Application components

5.4 Platforms

5.5 Security components

6 Migration to a private-cloud context

To perform a successful migration to a private cloud context we have analyzed the baseline performance of the current system, as well as the technologies used. Based on the client requirements, we have selected technologies and established a roadmap to perform this task.

6.1 Selected technologies

- Virtualization engine
- Storage arrangement
-

6.2 Architecture

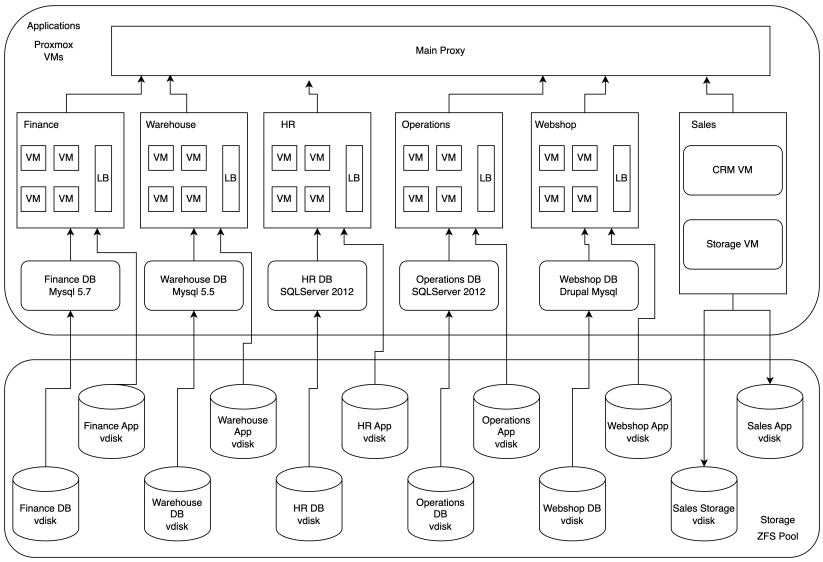

Figure 1: High Level Architecture 1

Note: We could use one VM and Docker to run most of the database instances as containers, making it easier to install or upgrade versions. Nevertheless, some of the tools are difficult to dockerize (SQLServer 2012) and, we want to have isolation between departments, preventing a single point of failure for all our database infrastructure.

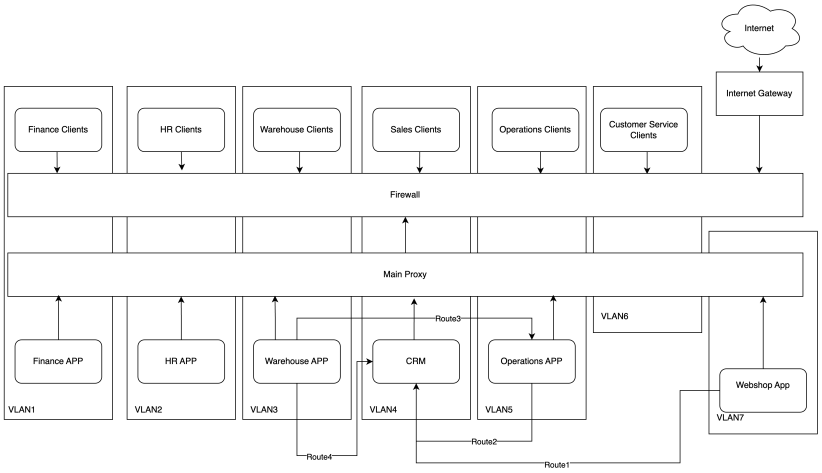

Figure 2: Network Architecture 2

We subdivide each department into its own network, virtually limiting the access of different clients to their corresponding systems. And only systems can have cross department routes to stablish communication for normal operation.

7 Roadmap

The technological transformation roadmap is divided in four phases:

- Assessment
- Design
- Execution
- Optimization

7.1 Assessment

This is the most critical step. We need to establish a baseline of performance, and cost of the current infrastructure. During this stage we are going to execute the following tasks:

- Review the current infrastructure in terms of performance, data requirements, availability and security. Establishing baseline metrics we then have some rough idea of the areas we want to pay attention, and at the end, the total impact that the migration had. We have some rough figures provided by the client, but these are not enough to actually guide development.
- Check the current applications, to define a migration approach, if they can be migrated, changed, or improved in a way that makes the transition process easier. Given the initial requirements, we can assume that the applications are not actively being developed, so modernization is not an approach we can consider at that level.
- Identify the level of effort per department/application/domain to make the migration, this metrics will guide the order of the execution. This is to reduce the impact of the migration in the business operation. Naturally there will be usage patterns more friendly to upgrade and intervention.
- Assess the budget and space constraints for new hardware, or to rent such services. A preliminary review points to renting server space as the most appropriate solution. To achieve the availability metrics required by the customer will require a great investment in provisioning, modernizing and validating the current server location. Not only in technological terms, but also: physical security, thermal management, internet bandwidth and energy provisioning.
- Spec the hardware after evaluating the usage pattern and metrics. We want to, not only get parity to the current hardware but have space to grow.

7.2 Design

In this step we perform the following tasks:

- Design the new architecture diagram with networking isolation between domains, dividing each department appropriately.
- Document which applications will be migrated to new technologies, or will be ported as is.
- Create document with detailing the dates and approximate duration of the migration each application. This is to be distributed to the organization so operation disruption is minimized
- Prepare contingency plans, and fallbacks to minimize disruption to the business if a migration fails. This could take the form of rollbacks, traffic duplication or A/B testing. Depending on the need of each department and the complexity of each application
- Create all the technical documentation regarding the implementation for the application migrations to new technologies.

7.3 Execution

In this step we perform the following tasks:

- Talk with server space providers, we need to stablish a contract and negotiate price and SLOs based on the server requirements
- Gather all the installation information, binaries, licenses and documentation for all applications.
- Configure a development environment to upload all artifacts such as configuration files and container images.
- Create Ansible configuration files for each application, to automate the installation and replication process.
- Develop scripts to automate application functionality and load testing. To determine if the configurations will be on par or better than the current infrastructure.
- Provision and install the private cloud management software, including the networking configuration
- Configure the storage
- Create the virtual machines for the base hosts
- Configure Prometheus to monitor the virtual machine installations
- Establish the network links between the different virtual domains
- Install all database servers
- Develop a migration process to replicate the current data into the new database servers.
- Review that data is up to parity with the legacy services
- Create a proxy getaway for the services, so we can redirect the traffic from the old services to the new ones.
- Deploy the new applications into the private cloud
- Check that the data replication is working
- Prepare the load shifting in sequence, and off hours
- Shift load in sequence, starting from the most isolated applications first, then the ones with the most dependencies.
  - Migrate finance and HR, since they are self-contained
  - Migrate operations
  - Migrate customer service
  - Migrate warehouse
  - Migrate webshop
  - Migrate sales

7.4 Optimization

- Measure the performance of the system using Prometheus metrics and create an assessment report of the migration.
- Create documentation regarding the new deployment process, provisioning and scaling.
- Assess potential improvement areas, and stablish follow up tasks if needed.
- Once we have the virtualized applications, we could start thinking into improving elasticity by enabling either ProxMox autoscaling (Which provides additional resources to VMs on the fly) or install an orchestrator that will create new virtual machines and load balance them.
- During optimization we also need to allocate time for training, and documenting the different processes such as: how to scale, deploy and provision new services.

8 Operation considerations

After preparing this technological transformation proposal it is important to remark that it will require an important level of effort to migrate the current infrastructure to a private cloud context. But we are not getting the entire benefit of a cloud application. It is to be determined if the applications will scale well horizontally, and if the load (particularly from the Webshop) increases, it will be costly to provision and configure more hardware. Moreover, it is still required to have skilled developers and infrastructure people dedicated to maintaining the health of the deployment.

9 Assessment of the (to-be) infrastructure

9.1 Improvements on Scalability

Elasticity and Resource Pooling via Virtualization: Use VMware vSphere and vSAN to enable dynamic resource pooling, which effectively distributes computing and storage resources.

Benefits:

- Automatic scaling based on workloads ensures optimum resource utilization.
- Reduces overprovisioning by dynamically allocating resources to virtual machines as needed.

Auto-Scaling Capabilities:

- Deploy Kubernetes for containerized applications (such as CRM and webshops). Kubernetes supports horizontal scalability, which allows more containers to be launched during peak loads.
- Incorporate Proxmox autoscaling for VMs, which dynamically allocates additional CPU, memory, and storage during peak demand.

Private Cloud with Centralized Management:

- Switch to a hyper-converged infrastructure, such as Dell VxRail or Nutanix NX-Series, for smooth scalability within the private cloud. The architecture can accommodate extra nodes without requiring considerable adjustment.

9.2 Improvements on Availability

High Availability Framework:

- Use VMware HA (High Availability) to ensure that services are restored fast following a breakdown. Combine with vMotion to relocate virtual machines without downtime during maintenance.
- Use distributed storage technologies such as Ceph to duplicate data across numerous nodes, maintaining availability even when hardware fails.

Redundant Architectures:

- Implement active-active clusters for key applications such as CRM and webshop. This guarantees ongoing service even if one node fails.
- Use load balancers (such as HAProxy) to distribute traffic across many instances of an application.

Backup and Disaster Recovery:

- Veeam Backup and Replication automates daily backups and provides fast recovery in the event of a failure.
- Introduce disaster recovery technologies such as Zerto for real-time VM replication, data integrity, and fast failovers.

Monitoring and Proactive Maintenance:

- Implement Prometheus and Grafana for infrastructure monitoring, which will generate real-time warnings for anomalies.
- Use Nagios to do automated health checks on servers and applications, which reduces manual involvement and potential human errors.

9.3 Improvements on Security

Enhanced Perimeter Security:

- Replace legacy firewalls with Next-Generation Firewalls (NGFW), such as Fortinet FortiGate or Palo Alto Networks, to improve threat detection and prevention.
- The advantages include application-aware filtering, intrusion prevention, and interaction with threat intelligence feeds.

Network Segmentation:

- Create VLANs to separate sensitive systems (finance, HR) from public-facing applications (webshop). This minimizes the attack surface.

Zero Trust Model:

- Implement a Zero Trust Architecture (ZTA) with identity providers such as Okta to enforce access restriction and MFA.
- Secure remote access with Fortinet VPN and extra MFA levels.

Regular Updates and Patch Management:

- Transition to latest operating systems such as Windows Server 2022 and Ubuntu 22.04 LTS.
- Use Ansible to automate updates and fixes across all servers and endpoints, ensuring that vulnerabilities are addressed quickly.

Centralized Log Management and Analytics:

- To gather, analyze, and store logs from firewalls, servers, and apps, use Splunk or the ELK Stack (Elasticsearch, Logstash, Kibana). This enables faster incident response and anomaly detection.

Backup Validation:

- Integrity checks should be included in Veeam Backup and Replication to verify that backups are free of corruption before restoring.

Integrating the CIA Triad into Infrastructure Strategy

Aspect	Proposed Improvements	Objective
Confidentiality	- Implement AES-256 encryption for data at rest and TLS 1.3 for data in transit. - Use Role-Based Access Control (RBAC) to enforce least privilege access policies.	Restrict access to authorized users and secure data in transit/storage.
Integrity	Enable backup integrity checks, SHA-256 hashing, and immutable infrastructure.	Ensure data accuracy and prevent unauthorized modifications.
Authenticity	Use digital certificates and code-signing to verify systems, users, and updates.	Validate the legitimacy of data sources and communications.
Non-Repudiation	Implement logging, auditing, and digital signatures to ensure accountability and prevent denial.	Track and prove actions or transactions without disputes.

Table 2: CIA Triad Improvements

References

1. Eurostat. (2023). *Electricity prices for non-household consumers - bi-annual data (from 2007 onwards)*. Retrieved November 16, 2023, from `https://ec.europa.eu/eurostat/databrowser/view/nrg_pc_205__custom_13581723/default/table?lang=en`
2. Mell, P. and Grance, T. (2011). *The NIST definition of cloud computing*. National Institute of Standards and Technology, Special Publication 800-145, Gaithersburg, MD. `https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf`