univention
be open.

Univention Corporate Server Architecture 0.0.1

Release 0.0.1

Univention GmbH

Jun 30, 2022

CONTENTS:

CHAPTER

ONE

INTRODUCTION

Welcome to the architecture documentation of Univention Corporate Server (UCS).

This document does not cover installation, the usage of UCS or parts of the product. For instructions about how to install and use UCS, see the UCS manual[1].

The document is released iteratively after each part is finished. The beginning is at the first, high level.

Your feedback is welcome and highly appreciated. If you have comments, suggestions, or criticism, please send your feedback[2] for document improvement.

1.1 Audience

This document is for consultants, administrators, solution architects, software developers and system engineers. It describes the technical architecture of UCS on three different detail levels.

The first, high level, *positions UCS in the known IT world* (page 3) and describes the *concepts* (page 7). This view helps readers to understand the principles of UCS. Chapters 2 and 3 assume you are familiar with information technology in general and that you have heard of computer network building blocks and software.

The second, medium level, is for administrators and solution architects. It covers the product components and the numerous services UCS offers to IT infrastructures. You read about the user facing product components and what services UCS runs. You learn what open source software contributes to the functionality of UCS and how it interoperates together.

Software developers and system engineers get an overview of the technical parts.

A general understanding of Linux operating systems for servers and IT administration are beneficial for understanding.

The third, low level is about the libraries, internal systems and storage. It describes the pieces a software developer and system engineer needs to know to contribute to UCS. General knowledge of software architecture and software engineering are helpful at this level.

1.2 Learning objectives

After reading this document you have a broad understanding of the UCS architecture. It equips consultants, administrators and solution architects to better plan their IT environment with UCS. It enables software developers and system engineers to quicker dive into software development for UCS.

[1] https://docs.software-univention.de/manual-5.0.html
[2] https://www.univention.com/feedback/?architecture=generic

CHAPTER

TWO

POSITIONING IN THE IT WORLD

To comprehend the architecture of Univention Corporate Server (UCS), it is important to understand the origin and where it is located in the world of information technology (IT).

2.1 Origin

UCS is a Linux distribution derived from Debian GNU/Linux[3]. Among others, it benefits from the strong software package manager, the high quality maintenance and the long-term stability as operating system for servers. Over the years, Debian has been and is a solid basis for UCS.

UCS is part of the open source family and has strong relations to important projects like for example Samba[4] and OpenLDAP[5].

2.1.1 History

Univention started UCS in 2002 as a collection of scripts that turn a Debian system into a Linux server that offers Windows domain functionality. The goal was to offer companies and organizations a standardized Linux server as alternative to Microsoft Windows Server that implements Microsoft's domain concept. Over the time it developed to an enterprise Linux distribution with maintenance cycles that better suited the needs of organizations.

2.1.2 Packages

On UCS software is managed in software packages. The packages on UCS use the deb file format. For more information on the deb file format, see the Wikipedia article deb (file format)[6] and Basics of the Debian package management system in the Debian FAQ[7].

UCS—like Debian—uses a package manager, which is a collection of software tools, to automate the process of installation, upgrade, configuration and removal of computer programs. Packages organize such computer programs on UCS. In UCS the package manager is the advanced package tool (APT). For more information about APT, see the Debian package management chapter in the Debian reference[8].

Univention distributes most packages from Debian GNU/Linux for the *amd64* and *all* architecture without changes for UCS. This includes the GNU/Linux kernel and over 98% of unchanged packages from the Debian project. Univention uses the default services from the Debian distribution and delivers custom configurations for UCS.

In the following circumstances, Univention builds and maintains derived packages:

- A later software version of a package is needed for UCS than Debian offers.
- Bug fixes or backports of a specific software are needed for a package.

[3] https://en.wikipedia.org/wiki/Debian
[4] https://en.wikipedia.org/wiki/Samba_(software)
[5] https://en.wikipedia.org/wiki/OpenLDAP
[6] https://en.wikipedia.org/wiki/Deb_(file_format)
[7] https://www.debian.org/doc/manuals/debian-faq/pkg-basics.en.html
[8] https://www.debian.org/doc/manuals/debian-reference/ch02.en.html

Additionally, Univention develops own software responsible for UCS functionality that is distributed as Debian package.

Nevertheless, UCS doesn't include packages from the Debian games section, because it would require a content rating for video games. Univention does not see added value in the distribution of video games with UCS for the product audience.

2.2 Identity management

The most important functional pillar of UCS is identity management.

Simplified, an IT environment consists of services and users. Services offer functionality. Users use functionality. Services can also behave as users when they use the functionality of another service. Users identify themselves against services to proof that they are eligible to use the functionality.

The identification is done with *user accounts* to represent users. User accounts typically have properties like for example username, password and email address. User accounts that digitally represent a person additionally have for example first name and last name.

Imagine a small IT environment with 20 persons and five systems. Without a central identity management, an administrator would have to maintain 20 user accounts on each of the five systems. The management effort sums up to 100 items. The number of items to manage is a linear function. The function's slope increases with the number of systems that need to know user accounts.

With a central identity management, one service holds the information about the user accounts. All other services have access to that information. An administrator only has to maintain the user accounts on one system. The maintenance effort for the user accounts does not anymore multiply with the number of systems that need to know the user accounts. The slope of this linear function is less steep.

Central identity management reduces the maintenance effort of user accounts for administrators.

UCS is a product for central identity management for user accounts, their permissions and the collection of user accounts in groups.

2.3 Infrastructure management

The second important functional pillar of UCS is IT infrastructure management.

IT infrastructure is a set of IT components like computer and networking hardware, various software and network components. It is the foundation of an organization's technology system and drives the organization's success.

UCS provides important infrastructure services to create an IT network infrastructure and connect IT components. For example UCS assigns addresses to computers and other network components through DHCP[9] and resolves hostnames through DNS[10], and much more. Administrators manage various IT components in their IT environment, like different kind of hosts, clients and printers.

2.4 Connection to the world around

As an operating system that offers many services, UCS interacts with its surrounding peers. Users access the functionality of UCS through the following ways:

Web Persons like administrators and also end users use HTTPS to access the web based UCS management system. In many cases other web-based services provided by other software products delivered through apps are also available through HTTPS.

[9] https://en.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol
[10] https://en.wikipedia.org/wiki/Domain_Name_System

Console Persons with more technical background and the appropriate permissions can access UCS through a console, either on a local terminal or through a remote secure shell (SSH)[11] session.

Service protocols As soon as users use any of the services that UCS offers, they access UCS through one of the service protocols. For example, a user's client requests and IP address through DHCP and later asks for the IP address of the print server through DNS.

As a central system offering identity and infrastructure management UCS has to use and offer numerous ways of connections.

[11] https://en.wikipedia.org/wiki/Secure_Shell

CHAPTER

THREE

CONCEPTS

Univention Corporate Server (UCS) unites numerous concepts to support administrators with their identity and infrastructure management tasks.

The following concepts explain how UCS uses them and are organized top-down:

1. *Domain concept* (page 7)
2. *Replication concept* (page 8)
3. *Role concept* (page 10)
4. *Permission concept* (page 11)
5. *Certificate infrastructure* (page 12)

3.1 Domain concept

The domain concept is the most important concept in an IT environment operated with Univention Corporate Server (UCS). The domain concept offers a way to centrally manage an IT environment where administrators can map their organization's structure to the IT environment.

Simplified, an IT environment consists of computer systems and users. Systems offer services that provide functionality. Users use functionality. A domain is a single trust context that groups one or more entities like computer systems or users. The domain offers special services called domain services to systems and users. The figure *Relationship of the systems, services and users in a domain* (page 8) shows the relationship between the actors systems, services and users.

A trust context uses roles, permissions and cryptographic certificates to ensure secure communication between the domain participants. Domain services and domain participants can rely on the shared trust context when secure and mutually authenticated communication is required.

One key participant in a domain is the identity, a digital representation for persons. An identity represents an account for a user in a domain. It holds information like for example username and password for login. Furthermore, it contains various data associated to the user like for example group memberships, permissions, and different attributes used by services.

User accounts are organized in groups and users can belong to multiple groups. User groups help administrators to apply permissions for domain services to users and are essential to the organization's structure to the domain administration.

All the objects in a domain need to be managed and organized. In a domain a central database called domain database registers all objects, like for example user identities, computer systems, printers and file shares. See figure *Central domain database with different objects* (page 9) for a graphical interpretation. The database stores the objects in a hierarchical tree-like structure. One or more central systems store the central database and are called domain node.

UCS is a system that operates the central database for the domain. UCS is the central platform that implements the domain concept and helps administrators to manage and organize the IT environment for their organization. For the distinct roles of UCS systems in a domain, see the *role concept* (page 10).

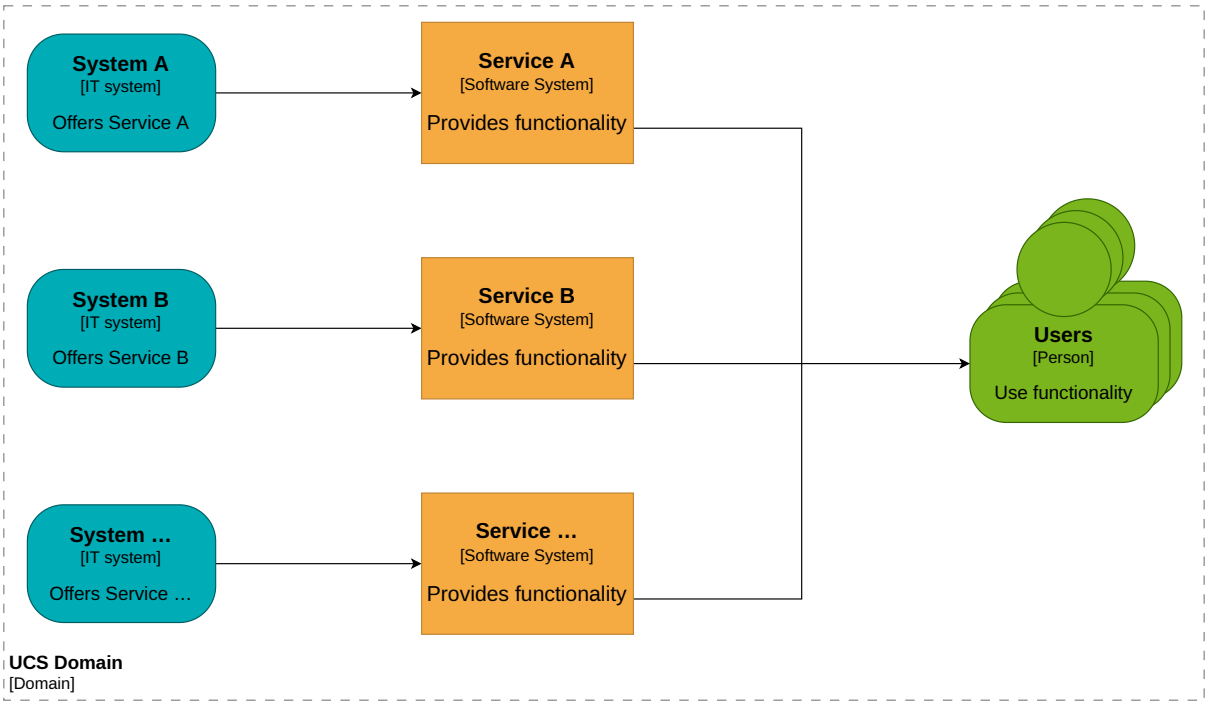

Fig. 3.1: Relationship of the *systems*, *services* and *users* in a domain

3.2 Replication concept

The replication concept ensures the availability and consistency of the central domain database and contributes to its scalability. It is necessary to keep the domain data synchronized across all domain nodes because more than one domain node can have a copy of the central database. For example domain nodes can get disconnected or need to shutdown for maintenance.

Univention Corporate Server (UCS) implements the replication concept. The first domain node in the domain has the following tasks:

- It writes domain object data to the database.
- It monitors changes to the database.
- It makes changes available to other domain nodes.

The other domain nodes have a read-only copy of the domain database.

The replication synchronizes a lot of data types. The following list names a few that domain nodes replicate and cannot cover all items:

- User identities
- Groups
- Policies
- Permissions
- Information about systems
- Information about printers
- Information about file shares

The domain replication in UCS also ensures that the affected UCS systems run follow-up actions once the changes are replicated. The actions can comprise of, for example, updates to configurations of services and making the changes available to the users.

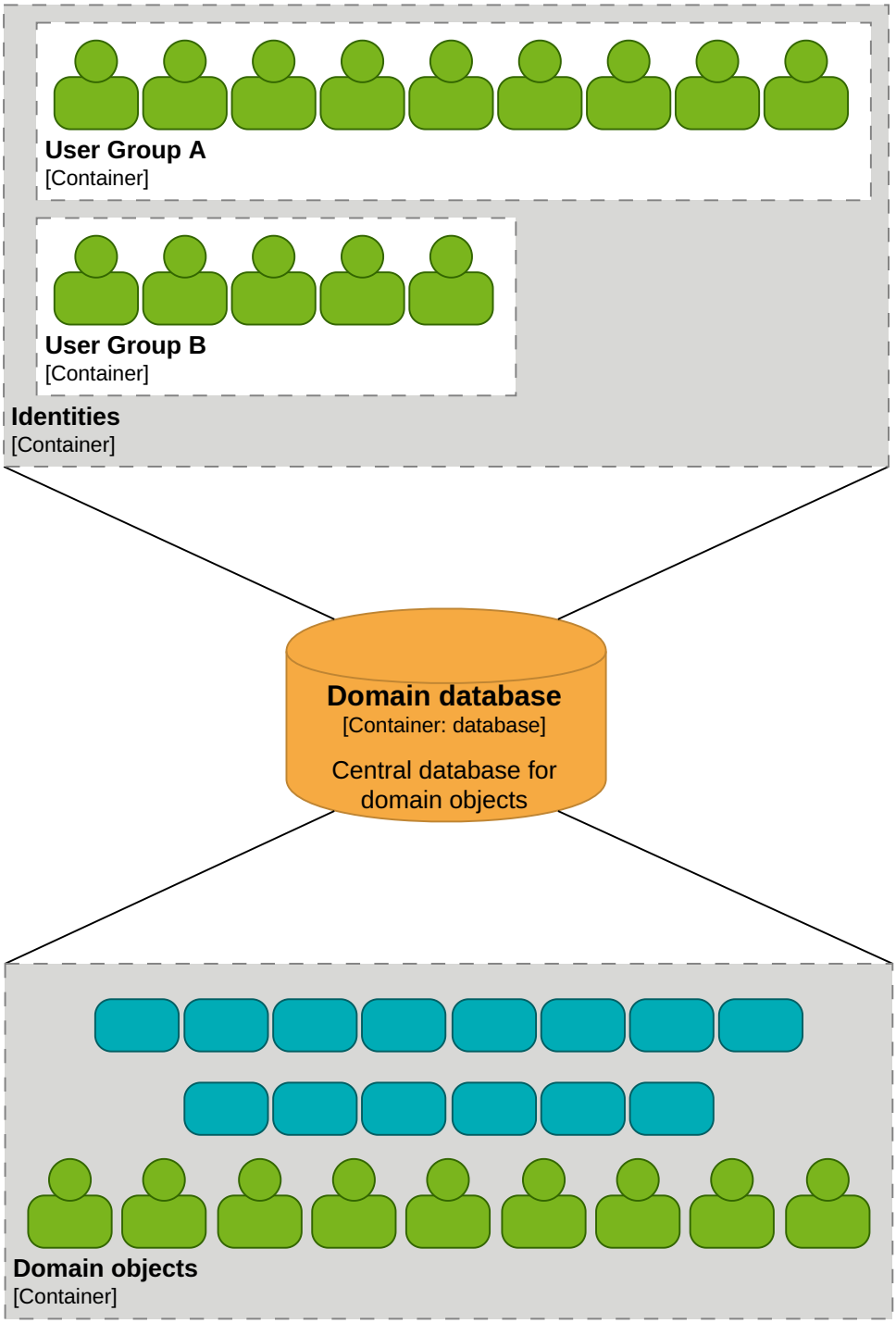

Fig. 3.2: Central domain database with different objects

3.3 Role concept

Univention Corporate Server (UCS) uses a role concept to assign different roles that comprise certain tasks to the systems in a domain.

3.3.1 Primary Directory Node

A UCS system with Primary Directory Node role is the first, the primary, domain node in a domain. It is the only system with write permissions to the central domain database and performs all write requests regarding data for the domain database. Only one system in the domain can have the Primary Directory Node role.

3.3.2 Backup Directory Node

A UCS system with the Backup Directory Node role has a complete read-only copy of the domain database, including security certificates. More than one UCS system can have the Backup Directory Node role. In case the Primary Directory Node is unavailable, recovery is impossible or needs too much time, an administrator can promote a UCS system in the role Backup Directory Node to a Primary Directory Node. The promotion cannot be reversed. For details on the promotion process, see Converting a Backup Directory Node backup to the new Primary Directory Node[12].

3.3.3 Replica Directory Node

UCS systems with the Replica Directory Node role have a complete read-only copy of the domain database. Administrators cannot promote Replicate Directory Nodes to the Primary Directory Node role or any other role unlike the Backup Directory Node.

A Replica Directory Node optionally allows selective replication, a form of data synchronization that replicates only a subset of the domain database. Selective replication in UCS helps with data minimization, domain protection and permission enforcement.

For example, imagine an organization with office locations in cities like Berlin and Bremen. Each location has a Replica Directory Node as domain node. The Replica Directory Nodes only replicate domain objects like users, groups and printers that are relevant for their respective location. They don't store objects assigned to other locations.

Replica Directory Nodes are ideally suited as dedicated systems for load intensive services with permanent read operations to the domain database because the read operations run locally instead of across the computer network.

3.3.4 Managed Node

UCS systems with the Managed Node role don't have any copy of the domain database. Services on Managed Nodes read domain information over the network from either the Primary Directory Node or from one of the Backup Directory Nodes.

[12] https://docs.software-univention.de/manual/5.0/en/domain-ldap/backup2master.html#domain-backup2master

3.3.5 Clients

Clients are systems in the domain's trust context. They don't have a special role regarding domain services as the other roles described before. In most cases they consume services offered by the domain or other systems.

UCS offers dedicated client roles for desktop systems like Ubuntu, other Linux desktops and macOS. UCS manages IP addresses and DNS entries for systems like network printers and routers with the *IP client* role.

For Microsoft Windows related systems, UCS offers the roles *Domain Trust Account*, *Windows Domaincontroller* and *Windows Workstation / Server*. For more information about the differences of these roles see UCS system roles[13].

3.4 Permission concept

The permission concept in Univention Corporate Server (UCS) specifies who can read and write domain data. Permissions apply to objects in the domain database like users and systems alike. Policies assign custom permissions to objects. UCS applies default permissions for systems and pre-defined users and groups.

3.4.1 System roles

UCS system roles imply certain permissions on domain data. Only the Primary Directory Node can write data to the domain database. All other system roles have read-only access. Nevertheless, other systems or users have write permissions for certain operations affecting themselves and they run them on the Primary Directory Node. For example, when a new UCS system joins the domain or an administrator installs an app write operations run on the Primary Directory Node.

3.4.2 Administrator and root

Some user accounts also have implicit permissions on domain data and systems. A UCS system knows two administrative user accounts: *Administrator* and *root*.

Administrator The user account *Administrator* is the first domain user and has all domain permissions. The *Administrator* user account has permission to join new systems to the domain and can work with all modules in the UCS management system. The account can only be defined once in the domain and must never be renamed.

The *Administrator* account is only defined once per domain during the installation of the Primary Directory Node. The account password is set during installation.

Think of *Administrator* as the primary administrative account for the UCS **domain**.

root The user account *root* is the superuser on the local UCS system and has the user ID of `0`. It has all permissions and is equivalent to the *root* account known from other GNU/Linux systems.

The *root* account is defined and the password is set during installation of every UCS system. The account is only for the local UCS system. On other UCS systems administrators should—of course—define different passwords for each *root* account.

Think of *root* as the primary administrative account for the **local** UCS system.

The *root* account has no permissions and is no valid account in the domain context. The account *root* must not be created as domain account.

[13] https://docs.software-univention.de/manual/5.0/en/domain-ldap/system-roles.html#system-roles

3.4.3 Domain users and admins

To simplify the assignment of certain user permissions, UCS has two default user groups in the domain that differ fundamentally: *Domain Users* and *Domain Admins*.

Domain Users UCS assigns every user to the user group *Domain Users* per default. The group identifies the user account as belonging to a person. The user account only has a minimal set of permissions in the domain.

For example, user accounts in the group can read the domain database, but cannot view password hashes. Additional apps in the domain like UCS@school and Fetchmail can alter read and write permissions for users and systems. User accounts in the *Domain Users* group also cannot log in to UCS systems for a remote shell by default. The UCS management system yields no modules for them either.

Domain Admins UCS creates one user account called *Administrator* during the installation of the first UCS system (Primary Directory Node) in a domain. It is the first user account and has all permissions for the domain. The *Administrator* user account is member of the *Domain Admins* group.

Users in *Domain Admins* group have all domain permissions just like the *Administrator* account. To join a UCS system to the domain, administrators need a user account that's member in the groups *Domain Admins* and *DC Backup Hosts*. For more information see Subsequent domain joins with univention-join[14].

3.4.4 Machine account

All systems part of the domain are actors in a domain like users. Each system has its own account in the domain database. The account is called *machine account*. Depending on the type of system they have different permission sets.

UCS systems can read data from the domain database with their machine account. Every machine account has assigned the following default permissions in the UCS domain:

- The UCS system can read all object information and password hashes for accounts from the domain database. Apps like UCS@school and Fetchmail limit the read permissions.
- The UCS system can write only information to the domain database that is associated with its account, for example the version of the installed UCS or other apps.

3.4.5 Policies

In addition to the permissions defined for system roles and pre-defined groups, UCS offers policies for more fine-grained control on administrative settings.

Policies are administrative settings to help administrators with infrastructure management that can be assigned to objects in the domain database. Policies use the inheritance principle as it is known from object oriented software programming. Inheritance allows to set policies to one object in the structured domain database. The policy then applies to all objects that are organized in the structure below.

3.5 Certificate infrastructure

The certificate infrastructure in a domain operated with Univention Corporate Server (UCS) ensures the trust context between all participants. The first domain node creates its own certificate authority (CA) for the domain. For more information see the Wikipedia article Certificate authority[15].

UCS uses Transport Layer Security (TLS). The UCS Primary Directory Node creates the CA on behalf of the domain during its installation and signs certificates for other systems that join the domain. All certificates have an expiration date. Backup Directory Nodes in the domain repeatedly pull all certificates from the Primary Domain Controller to allow administrators to promote one of them to a Primary Directory Node any time, if needed.

[14] https://docs.software-univention.de/manual/5.0/en/domain-ldap/domain-join.html#domain-ldap-subsequent-domain-joins-with-univention-join
[15] https://en.wikipedia.org/wiki/Certificate_authority

Services in the UCS domain also use the certificates created by UCS. Administrators can configure alternative certificates for end-user or internet facing services with certificates issued by third parties, for example Let's Encrypt[16].

The domain systems use the certificates for secure communication between each other over the computer network, for example for domain database replication and the web interface of the UCS management system. Communication clients need to know the public key of the domain's CA to validate the public key of the certificate.

[16] https://letsencrypt.org

INDEX