UBIQUITY
PRIVATE SERVER

Security Architecture White Paper

Last update: 2025/01/13

Contents

Abstract

This paper provides a description of the network architecture and security design of ASEM's UBIQUITY Private Server solution.

Audience

This document is aimed to network administrators, security auditors and decision makers to have a complete description of the security management and design to evaluate if UBIQUITY Private Server is compliant to their security standards and their use case scenarios.

Design Consideration

UBIQUITY Private Server's core task is to securely connect a client to remote devices through a private network or through the Internet (considered an insecure network). Thus, security is paramount on all design and implementation decisions, more than any other usability aspects.

Concepts

UBIQUITY Domain	It is a logical container that stores all the resources of a customer account: users, groups, devices and their configurations, folders, permission rules, and logs.

Component Architecture

Running server side

UBIQUITY Access Server	Consists of one server that manages the authentication of the UBIQUITY Manager/UBIQUITY Manager Tools and remote devices running the UBIQUITY Runtime.
UBIQUITY Relay Server	Consists of one (or optionally more) server that acts as a relay for an end-to-end connection between UBIQUITY Manager/UBIQUITY Manager Tools and remote devices running UBIQUITY Runtime.
UBIQUITY Web API	It is a server that exposes the API required for the UBIQUITY Manager/UBIQUITY Manager Tools to operate.

Running client side

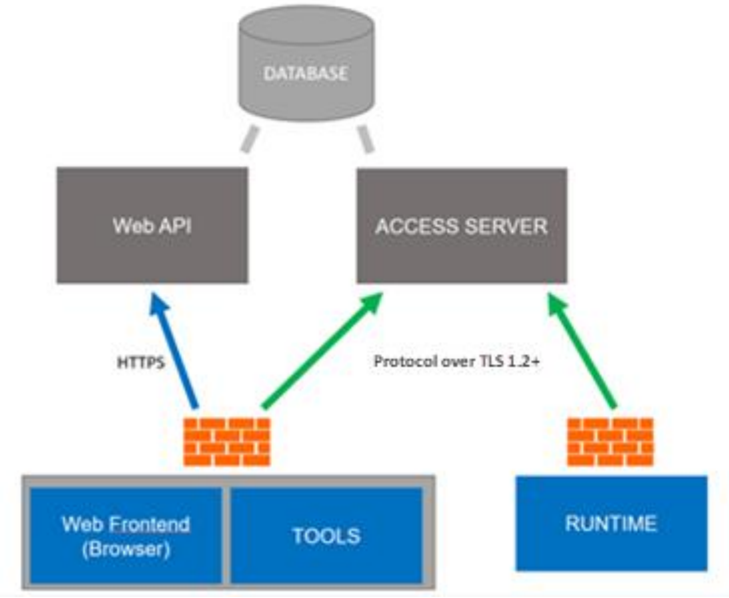
UBIQUITY Manager	Allow users to log into their UBIQUITY domain via Web Browser and connect to remote devices running UBIQUITY Runtime. Administrative users can also use the UBIQUITY Manager to manage security rules and device configuration. Also referred to as "client" in this document.
UBIQUITY Manager Tools	Applets started directly from the UBIQUITY Manager used to implement advanced features like VPN, advanced remote desktop, file transfer, remote process management, Remote serial/USB passthrough, etc. Also referred to as "client" in this document.
UBIQUITY Runtime	Runs on remote devices to enable remote access from clients. The Runtime is available as software to be installed on Windows and Linux industrial PCs or is natively embedded in ASEM's closed systems, such as UBIQUITY Routers and HMIs. The same security considerations apply.

Network and Protocol Design

To better understand the security architecture, it is important to have a brief understanding of the described components and how they work together.

The architecture can be divided in two parts, one supporting client connection and authentication and one supporting Client to Runtime connection.

UBIQUITY Access Server and Web API



Simplified network schema

The authentication to the UBIQUITY Domain is done by the Web API for the UBIQUITY Manager and Tools and by the Access Server for the Runtime. The UBIQUITY Domain information is stored in a database that is behind the Web API and Access Server.

All clients are assumed to be configured behind a firewall that only allows outgoing connections. The connection from clients to Access Server uses TLS 1.2+ with certificate authentication.

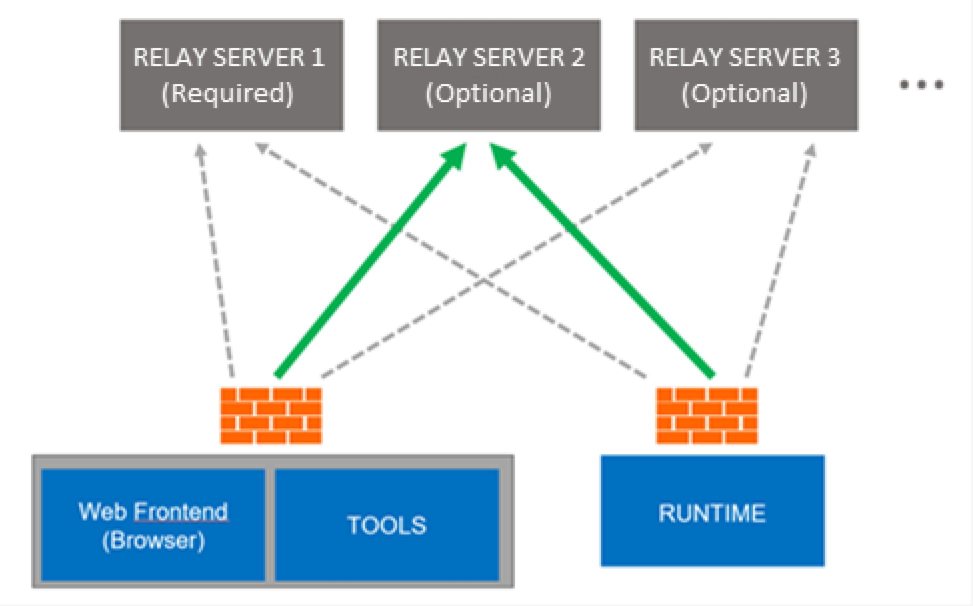
For connecting to the Access Server, clients can use the default TCP 443 outgoing port or can be configured to use port 80 or 5935 (TLS is still used of course), depending on what is best to comply with local IT policies. Clients automatically test available outgoing ports, but they can be configured for a fixed port.

Depending on customer requirements the Access Server can be kept in a private network or exposed as a public endpoint in a way that clients and Runtimes should be able to reach. Clients and Runtime communicates with this component by using a proprietary protocol on top of TLS; UBIQUITY Manager uses Secure Web Socket (WSS) protocol.

Web API is a ReST API that offers authentication/authorization and administration functions to Clients, such as administering folders, devices, users, groups, and so on or getting software updates download URLs. HTTPS on 443 TCP port is used for connecting to such service.

UBIQUITY Relay Server

When there is a remote access session between a Client and a Runtime, a Relay Server is used for data forwarding allowing both Clients and Runtimes to stay safe behind their firewalls as no incoming ports on their side are required to be open.


RELAY SERVER 1 (Required)
RELAY SERVER 2 (Optional)
RELAY SERVER 3 (Optional)
...
Web Frontend (Browser)
TOOLS
RUNTIME

Depending on customer requirements the Private Server can be kept in a private network or exposed as a public endpoint.

In case of multiple Relay Servers, the Client and the Runtime automatically choose the Relay Server to use from a pool of available list of servers list, provided dynamically by the Access Server.

In order to select a best Relay Server for a certain remote access session, both Client and Runtime perform a connection test to all Relay Servers and measure their respective network performances.
Both Client and Runtime results are then combined in order to select the best relay.
This automatic behavior can be disabled, and Clients can be configured to use a fixed Relay Server.

Transport Security

UBIQUITY Access Server Connection

TLS 1.2+ connections are used for the connection between clients and the Access Server. Both clients and Access Server set the allowed TLS cipher suites by following the security best practices. Used TLS algorithms (Key exchange, authentication, encoding and Message Authentication Code) can be easily verified by looking at the application logs, by using Wireshark or any tool which performs a TLS connection to the server.

Access Server use an SSL server certificate signed using SHA-256 with RSA by a well-known Certification Authority.

UBIQUITY Relay Server Connection

The End-to-end encryption securing remote access connections between Client and Runtime uses an AES-256 with a session key securely exchanged through a separate Access Server connection during the handshake phase. Since the Relay Server never participates to this handshake and is simply used after the session key has been exchanged, it cannot decode the incoming traffic and the connection is truly end-to-end secure.

The underlying transport is TCP (preferred for performance reasons) or TLS 1.2+ (as fallback for compatibility with firewalls requiring a TLS connection). Note that confidentiality is not guaranteed by TLS in this case, but by the upper-level AES-256 encapsulation. The protocol on top of the underlying transport is a proprietary protocol for Runtime and UBIQUITY Manager Tools.
UBIQUITY Manager, instead, always uses Secure Web Socket (WSS) protocol.

UBIQUITY Web API Connection

All clients use HTTPS for web APIs. Web servers use an SSL server certificate signed using SHA-256 with RSA by a well-known Certification Authority. Depending on customer requirements the Private Server can be kept in a private network or exposed as a public endpoint.

Remote access tools security

VPN

Once a Client is connected to a Runtime client, a VPN connection can be established depending on how the “VPN access” permission is given to a user on a given device.

The UBIQUITY VPN works at level 2 of the ISO/OSI protocol stack, i.e., it encapsulates Ethernet frames instead of IP packets. This is done for best compatibility with common industrial scenarios, where non-IP protocols or broadcast messages are used.

The VPN is implemented by installing a virtual Ethernet adapter on the Client PC.

UBIQUITY Runtime can intercept low level network traffic of selected physical interfaces and channel it to the Client’s virtual Ethernet adapter. For both the Client machine and the UBIQUITY Runtime device, it appears as if the Client machine is physically connected to the selected Runtime LAN.

Even if level 2 is below IP, by default the Runtime service automatically assigns a free IP to the Client virtual VPN adapter. This is done by convenience, since most useful protocols are IP based and thus ready to work. Moreover, IPs from the actual physical subnet(s) are used. No virtual IP subnets and consequent routing rules are created.

The Runtime periodically polls for existing devices on the network by sending ARP messages. It discovers “free” IPs that can be later assigned to VPN connections. This policy is handy but can be changed if a stricter and more controlled configuration is needed.
An IP pool can be configured on the device so the Runtime will only assign IPs coming from this pool. In this case no ARP discovery is performed.

Having the Client PC virtually connected to the physical device network is very powerful and handy, but it can be configured and limited in several ways to comply with ICT policies.

VPN firewall rules can be configured in the UBIQUITY domain to control what kind of traffic of a certain combination of device/sub-device/user/protocol can be remotely used. These rules can be obtained by configuring firewall rules across the domain hierarchy. Rules are hierarchical, per-user, per-resource, or per-resource group and can be limited to a certain remote MAC address, remote IPs, subnets, Ethernet or

IP protocols, in an ALLOW – DENY fashion.
The result set of rules are calculated by the server before a VPN connection starts and are enforced on both Client and Runtime.

The best practice security-wise is to only enable the protocols and reachable destination needed by a specific remote user or user group. This makes the VPN connection even more secure than an actual physical local connection, because in the latter case, the local PC firewall would be the only mechanism to limit traffic. In our case, the UBIQUITY infrastructure takes care of enforcing the security rules decided by the administrator.

File Transfer

Remote file operations (download, upload, rename, delete) are served through the UBIQUITY Service process. This process is running with local system privileges by default.
In any case, the UBIQUITY domain admin can enable or disable this operation for remote users depending on how the "File Transfer" permission is propagated to a certain device for a certain user.

Authentication

Authentication of Users

The users of UBIQUITY can log into the service with:

- Domain name
- Username
- Password

The UBIQUITY domain admins can enforce users to use a strong password by IEC 62443 standard. This enforces the following minimum requirements:

- Minimum 8 characters in length
- Contains at least 3 of the following items:
    - Uppercase Letters
    - Lowercase Letters
    - Numbers
    - Symbols
- Minimum 8 characters in length

Upon creating a new user by an admin, the user is required to change their password. It means that at any time, an admin can't login using a user's credentials without the user noticing (it would require setting a new password).

Two-Factor Authentication (2FA)

As an additional security measure, users can voluntarily – or enforced by the admin – use two-factor authentication (2FA).

UBIQUITY uses the Time-based One-Time Password algorithm (TOTP) adopted as Internet Engineering Task Force standard RFC 6238 and based on the HMAC-based One-time Password Algorithm published as an informational IETF RFC 4226.

It is compatible with Google Authenticator and most other two-factor authentication apps.

Authentication of Runtimes

When UBIQUITY Runtime connects to the Access Server for the first time, it obtains a signed identity file that contains the device UID in the UBIQUITY Domain. The certificate is used for authenticating devices to the server and relies to the operating system file system security. The certificate file is only accessible by elevated processes.

UBIQUITY Routers use an additional hardware feature that ensures device binding to a certain domain. In UBIQUITY Routers the UID is written in the hardware during the production stage in the factory and cannot be changed.
Once the Router is registered to a UBIQUITY Domain, it cannot be registered to another domain until the legitimate domain admin deregisters it from their domain. This is made possible by correlating the actual device identity to the hardware UID. This way, even if an attacker obtains physical access to the Router and performs a factory reset to reconfigure it, they won't be able to register it to their domain and thus use the Router for malicious remote access to the network.

Authentication of Clients

As mentioned, clients use TLS 1.2+ for connecting to Access Server. Access Server use an SSL server certificate signed by a well-known Certification Authority. Clients can verify the signature against the Certification Authority certificates installed in the system.

Operation Audit

The clients automatically record the operations performed on their domain resources by users (such as device removal, renaming, etc.) and send the information to the UBIQUITY domain. The audit log can be queried at any time by admins using UBIQUITY Manager and cannot be disabled or deleted, not even admins.

Each log holds:

- The user that performed the operation
- The operation code (such as rename of a device)
- The resource that was the object of the operation (e.g., a device)
- The timestamp
- A description

In more detail, the audit trail contains:

- Login/logout of users.
- All CRUD (create, rename, update, delete) operations performed on all domain resources:
  - Users
  - Groups
  - Permissions
  - Device
  - Configurations
- All remote access operations, with starting time and ending time.

In addition, the domain admin can *optionally* enable a full log of all operations performed during every remote access session including:

- Start/end times of remote-control sessions (view-only or interactive).
- Files transferred.
- Processes started and killed.
- Protocols, IPs and MACs contacted in a VPN session.

Authorization

UBIQUITY authorization model is similar to the Active Directory’s one.

The customer’s UBIQUITY domain contains all the domain resources: users, groups and devices.
The domain administrator can design the domain and its rules to map any kind of organizational structure. Specifically, new users and groups can be created to properly map the organization and make configuration easier and more scalable.
Policies and permission rules can be applied to single resources or folders to apply them in a hierarchical way.

Finally, the Web API and the Access Server will enforce types of operations (such as remote desktop or file transfer) that will be allowed for a certain user of a certain resource (such as remote device).

The UBIQUITY User Manual describes all the way authorization rules can be configured.

Cyber-attacks Countermeasures

Brute force detection

After a few unsuccessfully login attempts, Access Server completely blocks the incoming public IP address for a few minutes.

This measure, plus the password minimum requirements, makes brute force password discovery attempts ineffective.

Password management

Passwords in the UBIQUITY domain are not stored in clear text form. They are rather stored in hash form using a one-way cryptographic key derivation function and a per-user salt. The hash + salt pair ensures that even in case of data theft, an attacker cannot calculate the actual password from the hash by brute force using techniques such as rainbow tables.

Code Signing

UBIQUITY application binaries are signed with a private key. This ensures that users can always validate the authenticity and integrity of UBIQUITY applications.

Man-in-the-middle

Access Server, Web APIs and Client-Runtime connections all use end-to-end encryption (explained above) that make man-in-the-middle attacks not possible.

Hosting and Datacenter

Security Standards

In UBIQUITY Private Server, all the software components like the Access Server, Relay Servers, Web API and the Database must be maintained and take care by the customer.

We encourage the usage of multiple Relay Servers, hosted in heterogeneous and geographically spread locations to be statistically nearer either the Clients or the Runtime in order to lower latency. However, because of the special UBIQUITY network architecture explained above, they cannot cause any availability or security issue. Specifically:

- In case of multiple Relay Servers, Relay Servers are dynamically discovered. Even if a node is not available, clients will automatically switch to the others.
- Relay Servers are an intermediate hop of an end-to-end encrypted channel, explained above. A man-in-the-middle attack scenario is not possible.
- No data is stored on Relay Servers.

Backup

In UBIQUITY Private Server, the Backups must be take care by the customer. We suggest a SQL log backup to be done continuously with a Point In Time Restore of 7 days.

Software Updates

Updates of all UBIQUITY Private Server components are available on a regular basis.

Private Server infrastructure components such as Access Servers, Relay Servers and Web API can be provided upon request by the ASEM's Technical Support team that also manages the update of customer servers.

At the same time, updates of UBIQUITY Manager Tools, Runtime and firmware of devices embedding the Runtime can be downloaded and manually installed by the customer from the UBIQUITY website.

Data Breach Policy

A data breach generally refers to the unauthorized access and retrieval of information that may include corporate and/or personal data.

In the UBIQUITY Private Server context, data breach cannot affect customer's data given that the UBIQUITY Private Server software package is hosted directly by the customer.

In any case, the regulations across the various jurisdictions in which ASEM operates require ASEM to make reasonable security arrangements to protect the personal data that we possess or control, to prevent unauthorized access, collection, use, disclosure or similar risks.

Data breaches may be caused by employees, parties external to the organization, or computer system errors.

All members of staff have an obligation to report actual or potential data protection compliance failures. This allows us to:

- Investigate the failure and take remedial steps if necessary.
- Maintain a register of compliance failures.
- Notify the Supervisory Authority of any compliance failures that are material either in their own right or as part of a pattern of failures.

ASEM will notify any affected clients after becoming aware of a data breach. However, ASEM does not have to notify the data subjects if anonymized data is breached. Specifically, notifying data breach subjects is not required if the data controller has implemented techniques like encryption along with adequate technical and organizational protection measures to the personal data affected by the data breach.

Best Practices

- The UBIQUITY Domain should be configured to require strong user password per IEC 62443-3-3. Newly created domains have this option enabled by default. Refer to the User Manual for more information.
- Two-factor-authentication should be enforced by UBIQUITY Administrators for all users.
- In case of software installations on Windows machines, a firewall in the network (best if a hardware firewall) should be configured so that all connections from the Internet to the machine are blocked. Only one outgoing port should be used by UBIQUITY (TCP port 443, 80 or 5935) and kept open from the machine to the Internet
- Windows machines should only run controlled and safe software.
- The UBIQUITY software should be updated in case security improvements are released.
- Given the suggestions above, and given proper, static and controlled industrial environment, an antivirus software can be avoided.
- A strong administrator password change per IEC 62443-3-3 is enforced to register a Router to a domain. Please, keep the administrator password safe and do not share it with unauthorized personnel.
- UBIQUITY Routers can be connected to the Internet through its WAN port. UBIQUITY Routers don’t enable any service through that port and will only need an outgoing connection through to the configured outgoing port (TCP port 443, 80 or 5935). They basically don’t expose any surface to known attacks from the outside. We periodically test the latest version of the firmware stack against new kinds of attacks. However, for best security, an additional specialized firewall hardware would ensure the best protection from the outside.