WEM

SAFETY, SECURITY & GDPR

Rev. 1.8

Table of Contents

1 Legal

1.1 GDPR

The General Data Protection Regulation (AVG) applies since May 25th 2018. This means that from this date the same privacy legislation applies throughout the European Union (EU). The Personal Data Protection Act (Wbp) then no longer applies.

WEM (aka ZoomBIM or "WEM") has researched what consequences are of complying with this legislation. In line with this, a plan has been drawn up which indicates where risks lie and how to eliminate these risks with which ZoomBIM ultimately complies with the new legislation.

Characteristics against which we measure risk:

- Do we need the data?
- Is current access 'allowed/justified' (by current contract - role - etc.)
- Is current format in which we keep data, justifiable and safe?
- Does the current process create a risk on data leakage?
- What are generic processes we recognize (leave data on printer, lock screen when away from keyboard, keep door unlocked, Data storage procedures, ...)?
- Is data secure (enough) on location and format?

1.2 Data leakage

WEM recognizes the importance of keeping data safe and secure. Therefor there are several technical and organizational measures in place to prevent data from 'leaking'.

Should there accidentally be a suspicion or proof of data loss, there is a procedure in place to prevent further leakage and inform all stakeholders. All recordings are gathered and stored in one place and will be handled by one department and its current duty officer. This officer states the impact and will, according to protocol:

- take all necessary steps to prevent data from further loss
- report to all stakeholders, based on the impact of the recorded event
- take all necessary steps to prevent an event from ever happening again

1.3 Non-Disclosure Agreements

Should it be necessary to share certain data like privacy information, security information, etc., with others then we require a completed Non-Disclosure Agreement with all involved parties.

1.4 SLA

Service Level Agreements are generally available on three levels:

- WEM Runtime Infrastructure
- WEM Platform – WEM Platform SLA is provided by WEM. WEM will regularly update the platform to keep the components up to date and provide new features and bug fixes.
- WEM Project – WEM Project SLA's (SLA for projects developed using WEM technology) are generally offered by the WEM partner and are outside the scope of this document.

1.5 Data Processing Agreements

A Data Processing Agreement (DPA) is an agreement between a data owner (Responsible) and the data modifier (Modifier). A DPA usually will be offered at request of a WEM Customer or Partner.

All relations are invited to request a signed DPA. Should a party not have such an agreement of its own, then WEM is able to provide one.

Traditionally a DPA provides boundaries on what data the Modifier is allowed to handle.

1.6 Certification

ISO 27001

WEM obtained NEN-EN-ISO/IEC 27001:27001:2017 certification on the 13-5-2022. This certificate is valid until 13-5-2025. The ISO 27001 security management certification is the most widely used outside the United States. This certification consists of 133 controls and is applicable to the decor of the entire Information Security Management System. The certificate, Statement of Applicability and the audit report are available for inspection.

Cloud Provider ISO 27001/NEN 7510 Certification

For the WEM European data center clients, WEM's shared runtime infrastructure is situated with the Dutch data center provider CloudVPS, located at Delftsestraat 5B, 3013 AB in Rotterdam. CloudVPS is ISO 27001 (ISO/IEC 27001:2013) and NEN 7510 certified. CloudVPS has implemented the 2005 version in 2012, the ISO 27001:2005. In 2015, CloudVPS has been certified again based upon the 2013 version: ISO27001:2013. The certificate, Statement of Applicability and the audit report are available for inspection.

NEN 7510

The Dutch healthcare sector modifies and saves important medical and patient information on patients. To ensure that medical information is kept save, the NEN (the Dutch Standardization Institute) created the NEN 7510 security standard. CloudVPS has the NEN 7510 implemented at the same time as the ISO 27001 and was audited accordingly. CloudVPS has implemented technical and organizational measures to ensure the integrity of its patient data and pass the audit. With the use of firewalls and technical and physical separation of networks, it is not possible to inappropriately access data.

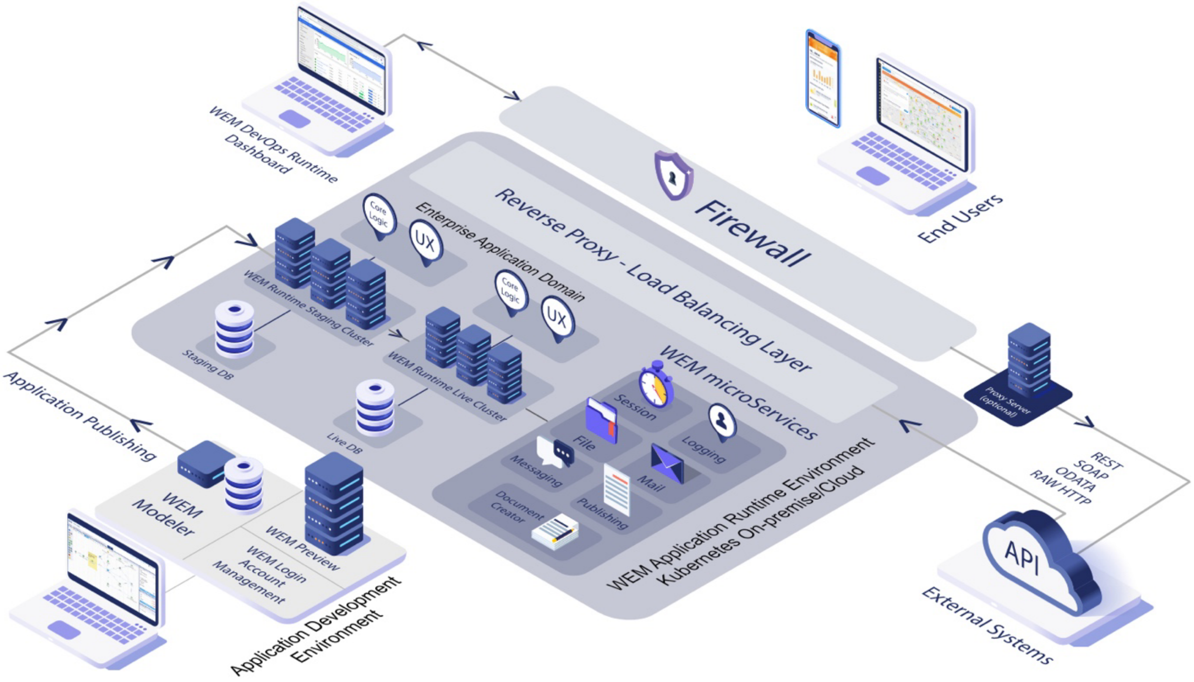
Regarding countries outside The Netherlands, WEM will only contract hosts that are certified.

2 Software

Regarding access to data and the technology of WEM itself, there are several measures in place as part of the WEM platform and the infrastructure setup. The WEM platform has 2 distinct elements:

1) Modeler (development)
2) Runtime (application platform)

While there is one modeler environment, there are multiple runtime environments, both on premise (or private cloud) and on our shared cloud environments. Every WEM application is linked to at least one staging runtime environment and one live runtime environment to enable a DTAP street.



2.1 WEM Modeler

WEM applications are built in the WEM modeler. From the modeler the application is published to the run-time environments. The publication happens in real time to the Preview environment and on request to the Staging and Live environments. The WEM modeler will never have direct access to the application data stored in the databases of Staging or of the Live environment.

2.1.1 Authentication

User password in the WEM modeler are hashed using PBKDF2 with both salt and pepper (the pepper is not stored in the database). User authentication is logged in an audit table. The modeler contains tight role-based and context-based access control for its users. Two factor authentication using SMS or an authentication app is also offered.

2.1.2 Sensitive information

Sensitive information stored in the modeler (like API secret keys and X.509 client certificates) are encrypted using a private key based on a X.509 certificate that is only available in the modeler environment. When such information is published to a runtime environment, then these resources are re-encrypted with the unique public key of the target runtime environment. Only the target runtime environment can decrypt it with the corresponding private key.

2.1.3 Publishing

The WEM modeler creates applications that can be published to a WEM runtime. Packages are sent to the runtime environment over a secure connection and are signed using a private key based on an X.509 certificate that is only accessible by the production modeler publisher service. The publishing mechanism contains measures to prevent replay attacks.

2.1.4 Preview

The preview environment is a runtime environment that is tightly coupled with the modeler and provides a real time feedback loop while developing an application. Access control to the preview is integrated with the modeler user accounts.

2.2 WEM runtime

The WEM runtime runs applications and can run on multiple cloud environments. The WEM runtime contains multiple components that can all run in a high-availability cluster to distribute load and minimize downtime.

2.2.1 HTTPS

HTTPS (TLS) is enabled for every WEM application by default. The list of allowed cipher suites can be configured per application. However, WEM does not support SSL3 and TLS1.0 anymore.

2.2.2 Runtime API security

The runtime components communicate through internal APIs. All calls to internal API endpoints are signed with a shared key. All API calls have measures to prevent replay attacks. APIs that are public facing require authentication based on X.509 certificates.

2.2.3 IP based access control

The runtime environment provides IP based access control. This can be fine-tuned for specific endpoints so that for instance an external application can only access a single API endpoint.

2.2.4 Application authentication and authorization

The authentication and authorization of WEM applications is implemented in the WEM project itself and can be completely customized. However, there exists a template project that follows common best practices:

- Passwords require a minimum strength (defined in entropy).
- Passwords are stored as a PBKDF2 hash with a private salt and a global pepper.
- Proper mechanisms to reset a password.
- An audit table that logs all login attempts.
- Access control is both role-based and context based.

While the template project contains a basic authentication implementation, WEM supports a wide range of custom implementations, including:

- Basic HTTP username/password (with possibility to configure the amount of attempts)
- SAML 2.0
- OAuth 2.0
- Two-factor authentication (e.g. with Email or SMS, using your own external provider)
- Challenge/response
- Captcha
- IP/network firewalling

WEM supports single sign on (SSO) for the SAML and OAuth protocols for all applications created with WEM. This includes integration with Active Directory through ADFS.

2.2.5 SOAP, Rest and OData webservices

WEM applications can provide custom API endpoints using SOAP, REST or OData. These endpoints can use the following built-in security measures out of the box:

- Required TLS connection
- IP access restrictions
- OAuth 2.0 authentication
- Authentication based on X.509 client certificates
- Basic HTTP authentication

Other security measures (custom authentication, hashing, encryption, replay attack preventions, etc.) can be implemented using standard WEM functionality.

2.2.6 Monitoring and logging

The WEM runtime monitors and logs the following information:

- The uptime of all its components.
- All inbound HTTP requests
- All outbound HTTP requests
- All outbound email messages (optional)
- All application errors
- Custom application logging.

2.2.7 Storage of sensitive data

Sensitive data at rest can be hashed or encrypted before storage. WEM support includes AES encryption, Sha-256, HmacSha256, PBKDF2 and MD5.

2.2.8 Injection prevention

Injection flaws, such as SQL, OS, XXE, and LDAP injection occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.

WEM prevents injection flaws by treating all input as possible hostile. This includes query string parameters, HTTP headers and cookies. All input will be parsed and converted to their native data types (numbers, dates, booleans, enum values, etc.) before usage.

All serialized data (e.g. XML, JSON, CSV) is validated during deserialization.

SQL code is never generated by string concatenation but will always be constructed by building an abstract syntax tree. All SQL commands are parameterized.

2.2.9 Cross-site scripting (XSS)

XSS flaws occur whenever an application includes untrusted data in a new web page without proper validation or escaping. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.

The WEM runtime will escape all user generated content by default (whether this is part of the HTML, URL, XML, javascript or JSON). HTML blobs (rich text user input) will be sanitized using a whitelist before usage. Certain mime types such as SVG can be blocked from upload.

2.2.10 Broken Access Control

Restrictions on what authenticated users are allowed to do are not properly enforced. Attackers can exploit these flaws to access unauthorized functionality and/or data, such as access other users' accounts, view sensitive files, modify other users' data, change access rights, etc.

WEM provides "what you see is what you get" security that only allows the user to perform an action if it is enabled and visible in the UI. WEM keeps track of a "blueprint" that describes exactly what the user is allowed to do at a specific moment and will validate each HTTP request against this blueprint.

2.2.11 Cross-Site Request Forgery (CSRF)

A CSRF attack forces a logged-on victim's browser to send a forged HTTP request, including the victim's session cookie and any other automatically included authentication information, to a vulnerable web application. Such an attack allows the attacker to force a victim's browser to generate requests the vulnerable application thinks are legitimate requests from the victim.

WEM uses the synchronizer token pattern (STP) to mitigate CSRF attacks. After each request, the server sends a unique token that must be used for the subsequent request.

2.2.12 HTTP headers and session cookies

Session cookies are generated by a cryptographically secure random number generator. Session cookies contain the "__Host" prefix and are provided with the "secure" and "HttpOnly" attributes.

Additional HTTP headers may be enabled for improved security:

- Content-Security-Policy
- Refer-Policy
- Strict-Transport-Security
- X-Content-Type-Options

- X-Frame-Options
- X-XSS-Protection (although this feature is deprecated by all modern browsers)

2.2.13 The use of third-party components

Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts.

We maintain a list of all third-party components used by WEM. Vulnerabilities of these components are checked against the CVE database. We regularly check for available security updates and patches.

2.2.14 HTTP request validation

The WEM runtime validates every HTTP request (headers, cookies, CSRF token, etc.) and if the request is not correct, then WEM silently ignores the request and sends a HTTP 200 response. This minimizes the feedback that potential hackers get from the system.

2.3 Development process

2.3.1 Code review

Each feature is developed in isolation on a separate branch. Before any code is committed to the main branch, the code will be reviewed by team members for:

- Quality – does it conform our code and quality guidelines?
- Stability – how will an incident in the component affect the stability of the system as a whole? Are there mechanisms to prevent escalation? Does it introduce a single point of failure?
- Regressions – is it backwards compatible? Does it adversely affect existing features?
- Security – does it create a new attack vector? Are all APIs secure? Is the input validated?
- Performance – are there pathological cases? Does it have a negative effect on the performance of existing systems?
- Scalability – will the component scale? Can it run in a load-balanced cluster?
- Logging – will incidents be detected and logged? Will this provide enough information to investigate the root cause of the incident?

2.3.2 Automated testing

New release candidates are automatically built and deployed to our test environment by our CI/CD setup. Release candidates are guarded by automated tests before going to production. These tests consist of:

- Several thousand unit-tests.
- Several thousand integrated regression tests.

2.3.3 Release automation

New releases are automatically distributed by our CI/CD setup. Updates are packaged and pushed to remote runtime environments over a secure connection. Packages are signed by our build server to ensure the authenticity of the update. Most of the time, new releases are installed with zero downtime.

The option to push a new release to production is restricted to three WEM employees.

2.4 Privileged access management

Privileged access management in WEM is done on 3 different levels. For each level the responsibility can be different. In addition to that, the privileged access management for a shared runtime can be different than a private runtime. In the below table this is explained per level.

Level	Responsible	Explanation
Application	Project owner	Role based access in the application can be custom created per the requirements of the customer.
Modeler	Workspace/Project owner	WEM has several roles that can be assigned to a user. More information about this topic can be found in the documentation: User Roles. The owner of the workspace or project can assign the user role of every connected account.
Shared Runtime	WEM	The shared runtime and databases can only be accessed by WEM via a jump host. Which in turn is only accessible via a VPN. Only a small number of vetted

		and experienced WEM employees is allowed to access the database.
Private runtime	Runtime owner	In a private runtime the owner of that runtime is responsible for defining the access management. Depending on the support required access could be given to WEM employees or WEM could be completely prohibited from accessing the runtime and databases.

3 Project owner responsibilities

WEM allows anybody to create a custom application. WEM allows the project owner to decide which data is visible in which situation. This flexibility means the project owner is also responsible for part of the security of the application. In situations where the project owner of the application is part of a customer or partner, this customer or partner is responsible for making sure the application is built correctly.

3.1 Access control

The first on the list of the OWASP top 10 is Broken access control. The OWASP description:

Access control enforces policy such that users cannot act outside of their intended permissions. Failures typically lead to unauthorized information disclosure, modification, or destruction of all data or performing a business function outside the user's limits.

The WEM project owner has full control over roll-based access control and is therefore responsible for making sure this is implemented correctly. This means using the "Visible when" and "Allow access when" feature correctly. WEM is not responsible for data leakage resulting from broken access control caused by the user. This includes the logging, measuring and handling of (un)successful login attempts.

3.2 IP based access control

The modeler allows the project owner to black- or whitelist IP addresses. This can be set for the entire application or per portal, runtime or even specific endpoint. Explanation of how to use this feature can be found in the documentation: IP Restrictions. Any blacklisted IP is stopped in the application server before any code is being executed.

In addition to that an expression called IpAddress() can be used to return the IP address of any given request. This allows for extra custom logic in the flowcharts.

3.3 HTTP headers and Ciphers

WEM applications are by default not set to the highest safety standard regarding HTTP headers and Ciphers. This is done on purpose to allow WEM to use widgets and allow connections with legacy systems. This will also mean that a WEM application might not adhere to every company's safety standard by default. A lot of these safety standard could result in some features not working. It is possible to change the headers and ciphers to almost every desired configuration. Due to the possible impact on an application, it is the project owner's responsibility to ask WEM to set the headers and ciphers to their specification. A feature is currently in development allowing everybody to change this without the help of WEM.

3.4 Widgets and Custom script

WEM allows the use of widgets and custom script built with Javascript, CSS and HTML. WEM has made several widgets available via the global library for which WEM is responsible. The WEM project owner is responsible for making sure any non-global widget or custom script is built in a secure manner.

3.5 Custom design template

WEM allows the creating of custom design templates via the WMT tool. Such a design template may contain custom scripts or other elements that could contain vulnerabilities. The project owner is responsible for the security of a custom design template.

3.6 Clear session

WEM has a built-in feature called the Clear session node. The project owner of the application is responsible for making sure this is used at the correct time to prevent unauthorized access to the application.

3.7 Webservices and oData

WEM allows exposing webservices. These webservices can be secured in several ways (ie: username & password, certificates). In addition to that the webservice can be encrypted. The project owner of the application is responsible for making sure the webservice is secured correctly according to their own standards.

4 Penetration test

WEM has a "Technical vulnerability management policy" according to the ISO27001 Annex A.12.6. This policy is reviewed at least once a year and audited during the ISO27001 audit. In this policy, a chapter about penetration tests is included. Every year a minimum of 2 penetration tests needs to be performed on the runtime. Every time an application running on a runtime is tested, the runtime itself is tested as well. Since every customer is running on the same runtime, a test performed by a customer on their application(s), is counted as a performed penetration test on the WEM runtime.

4.1 Performing penetration tests

All customers are allowed to perform a penetration test if they comply with the below rules of engagement. Some of these rules are only applicable when the application is running on a shared runtime.

Rule	**Applicability**
The penetration test is announced at least a month in advance.	Shared and Dedicated
WEM is provided with the start and end date of the penetration test	Shared and Dedicated
The IP range of which the penetration tests is launched will be communicated in advance	Shared and Dedicated
The URL(s) that will be targeted during the attack are communicated in advance	Shared and Dedicated
Attacks will not be aimed at any Live runtime URL	Shared only
The following will not be part of a penetration test: DNS server attacks DDoS attacks	Shared only
WEM has provided written approval of the penetration test before it starts	Shared and Dedicated
When an attack vector has proved successful, WEM will be notified and given the necessary steps to reproduce the attack.	Shared and Dedicated
All information related to this penetration test will be treated as confidential and will not be shared with third parties without consent given by WEM.	Shared and Dedicated
WEM is provided with an anonymized copy of the report that WEM is allowed to share with third parties on its own discretion	Shared and Dedicated

4.2 Analyses

Typically, any findings are scored using the Common Vulnerability Scoring System. This rating includes a Qualitative rating (None – Low – Medium – High – Critical) that is derived from a CVSS score. For more information: https://www.first.org/cvss/specification-document#Qualitative-Severity-Rating-Scale

Customers are encouraged to report any vulnerabilities with a rating of High or Critical (CVS score 7.0 or higher) immediately after finding them. Any findings with a lower rating can wait until the security company has delivered the report. If a customer reports a vulnerability of high or critical, WEM will treat this as the most important issue in the company and will immediately start the analyses.

Analyses are performed by a team of people consisting of the CTO, COO and others depending on the expertise needed. Usually these are senior WEM developers and/or senior consultants from WEM, Partners or the Customer. Lead time of the analyses is depended on the severity of the vulnerabilities reported. High or Critical should be analyzed in no more than 3 days. Medium and lower will have lead time of maximum 3 weeks.

The process is always the same:

1. The report and all findings are added to the WEM ISMS (Information security management system)
2. The issues are reproduced in a local environment
3. If the issue can be reproduced it is analyzed (with high or critical findings an estimation of the lead time is provided after the first analyses)
4. The vulnerability is classified (see next chapter)
5. The classification is explained to the partner and customer
6. If needed WEM provides explanation or help in fixing vulnerabilities where WEM does not accept responsibility (see chapter 4)
7. If accepted as a vulnerability a fix is planned. Timeline depending on severity of the vulnerability and complexity of the fix
8. The fix is released and communicated to partner and customer

4.3 Classification

Any vulnerability is analyzed and then classified. This classification is the basis of the feedback to the security company and determines the next steps. All classifications are accompanied with a justification in writing.

Classification	Explanation
Accepted	WEM agrees that this is a vulnerability and it is determined as a bug in the WEM runtime and will need to be fixed as soon as possible.
Not accepted	WEM does not agree that this is a vulnerability.
Not applicable	WEM agrees that this could be a vulnerability but it is not applicable in the context of the application or runtime.
Modeler setting	This vulnerability is caused by incorrect use of the modeler and can be fixed by the partner/customer. If needed with help from WEM.
Infrastructure setting	This vulnerability is due to infrastructure settings. Al companies have different guidelines for this. These settings can be set for every portal. Examples are HTTP headers and Cyphers. These can be prevented by checking these before the penetration test.
Third party	The vulnerability is caused by a third party. This could be a widget, custom script, or integration with a third party. This is outside of the responsibility of WEM.

4.4 Possible reported vulnerabilities

Every year the WEM runtime is tested at least 5-10 times by various customers. Because all these customers can have very different applications, the penetration tests can be very different. This ensures that all features of the runtime are tested extensively. Over the years the WEM runtime has been tested by several specialized security companies on behalf of (enterprise) customers in various industries. These include banks, national governments, insurance, telecom, logistics and many more.

Over the years WEM has received multiple vulnerabilities. None of those has been a vulnerability classified as Critical. A small number has been classified as High, of which only one (CKeditor) in the last 3 years has been Accepted by WEM as a vulnerability and promptly fixed within 2 days in a weekend. In preparation of the penetration test several possible findings are mentioned below. Most of these can be prevented by following the suggestions.

4.4.1 jQuery

The runtime uses jQuery for some of it's components. The version that is used, is defined in the design template. It can either use a specific version, or automatically use the latest version. Some of the older design templates are still using an older version of jQuery. It is wise to check the version your design template is using and make sure it is using 3.5.1 or later.

4.4.2 Bootstrap

Some of the runtime code is based on 3.4.1. This is an older version of bootstrap which could be mentioned as a finding. Bootstrap 3.4.1 is the latest version of the bootstrap 3.x branch. Bootstrap version 3.4.1 was released to address some security vulnerabilities. No new vulnerabilities were found since this version. We track the CVE database for vulnerabilities, and we will provide a patch if any are found.

Additional notes:

- Bootstrap 3 and 4 co-exist. Version 3.4.1 was released over a year **after** version 4.0 was released.
- The bootstrap Javascript code is very basic and therefore easy to maintain. Its primary function is to add basic behaviour to UI components such as panels, tooltips and dropdown-menus.

4.4.3 SV Token

The runtime uses a query string token (the "sv" token) that encapsulates the current state of the application in the browser. This token is encrypted and signed to prevent user tampering. This token is bound to the current user session and does not contain sensitive information. If an attacker gets hold of this token, it is harmless.

4.4.4 CKeditor

The runtime currently uses an older version of the CKEditor. This version is known to be subject to some XSS vulnerabilities when it is given corrupted HTML data. WEM always validates the HTML data that is passed to the CKeditor by using a strict whitelist of html elements and attributes. This validation and if needed sanitation is performed on saving of the data. This has been thoroughly tested and confirmed by multiple PEN testers.

4.4.5 Server hardening

The WEM runtime adds the "Server" HTTP header to the response message with the value "NGINX". NGINX serves as a reverse proxy in front of the actual application servers. The actual webserver and technology stack is hidden from the user.

4.4.6 Unrestricted file upload

WEM does not perform any validations on the type of file uploaded to the system unless specified in the properties of a file field. This means that WEM allows malicious files to be uploaded to the system. These files are stored in a binary format on the filestore and can therefore do no harm to the system. But it is possible to distribute malicious files through the system. To prevent this a property is provided to restrict the upload to a single type of file. If multiple type of files are allowed a whitelisting has to be created in the system which validates the file using the “mimetype” expression.

4.4.7 Unrestricted file download

A file field has a required property called “Allow access when”. This property allows the developer of the application to restrict to possibility that a file can be downloaded only if certain conditions are met. A commonly used expression is one that validates if the file is being downloaded by a logged in user. But these expressions can be much more elaborate than that. For instance, only users with a certain role can download the file.

During the development of the application, we see a lot of people that use only the expression “True” in the “Allow access when” property. This means anybody with the right URL can access the file, leading to a situation where an attacker can potentially guess the file URL. It is strongly advised to check this property for all file fields before the penetration test has started.

4.4.8 Virus scanner

As mentioned above, any file uploaded to the WEM runtime cannot be executed on the runtime servers itself and is therefore harmless. But it is still possible to distribute malicious files through a WEM application. Therefore, a Virus scanner pod is in development that can scan any file uploaded to the runtime. This feature is expected to be release to the Kubernetes runtime in Q2 2023.

4.4.9 Headers, TLS and Cyphers

Headers, TLS and Cyphers are partly depended on guidelines of an organization. Therefore, they are customizable per runtime. To strict could mean certain features, like widgets, legacy integrations, and external CSS, do not work. In the past WEM has used an open policy where the settings are not set very strict. WEM will try using the latest version but if they do not work an older version is accepted. This means that those settings might not adhere to your company’s standard. This is mostly the case with applications running on the Shared European runtime. For Kubernetes based runtime WEM has changed to a as secure as possible policy where less secure options should be specifically set.

If you application is running on the shared runtime it is advisable to use https://www.ssllabs.com/ssltest/ before the penetration test starts. The required setting can then be changed to your company's policy.

4.4.10 Widgets and Third party integrations

Widgets and Third party integrations are regularly the cause of vulnerabilities. WEM is responsible for any widgets that are provided through the global widget library. WEM does not take any responsibility for any other widget or integration.

4.4.11 Session timeout

Experience has thought us that the session timeout is very company and sometimes even applications specific. The WEM runtime has a default session timeout of 240 min. Admittedly this is long, so this will be changed in the near future. But this is also an infrastructure setting that can be easily configured per portal before the penetration test starts.

4.4.12 Custom Error handlers

Sometimes an application is built correctly and impossible to enter without proper authentication. In most penetration test this will be tested by trying to navigate to a specific url path and see the result. This could be for instance wem.io/accounts or a specific URL of a file in the filesystem. In some applications this could for example result in a 404 not found or an Unauthorized error. If this is mentioned, it is most of the time classified as a medium or low vulnerability. This is due to Custom error handlers not being set. It is therefore very easy to prevent by making sure all the Custom error handlers have flowcharts that handle these situations. These can be found in the portal settings. Make sure these are set for all portals in use.

4.4.13 User login

User login can have multiple issues that could result in a vulnerability being reported. These things have only been reported in situations where the login functionality is built in WEM. No applications with Single sign on have had these situations. Unfortunately, these are not yet fixed in the standard Authentication module (at the moment of writing this document) and it's therefore wise to check for them before the pen test starts.

4.4.13.1 Unlimited login attempts

Whenever a user tries to login with the wrong credentials an error is given. But often there is no limited to the number of wrong attempts the application allows. In theory this means a password can be brute forced. This can be mitigated in 2 ways. Either the user account is blocked after X attempts (mostly between 3 and 10) or the user has to wait for X seconds (mostly 1 minute) before they can try again.

4.4.13.2 Password strength

WEM has an expression called Passwordstrength. This allows you to measure the strength of any given password and returns a value between 1 and 5. In the authentication module the minimum requirement for a password strength is 3. For some penetration testers this is not enough.

Another thing that could be reported is that the password does not require specific rules like Capital letters, numbers of special characters. The passwordstrength only checks the strength and not the contents of the password. This is a little bit more work to mitigate because you would need to use regular expressions to measure this.

4.4.13.3 Existing accounts check

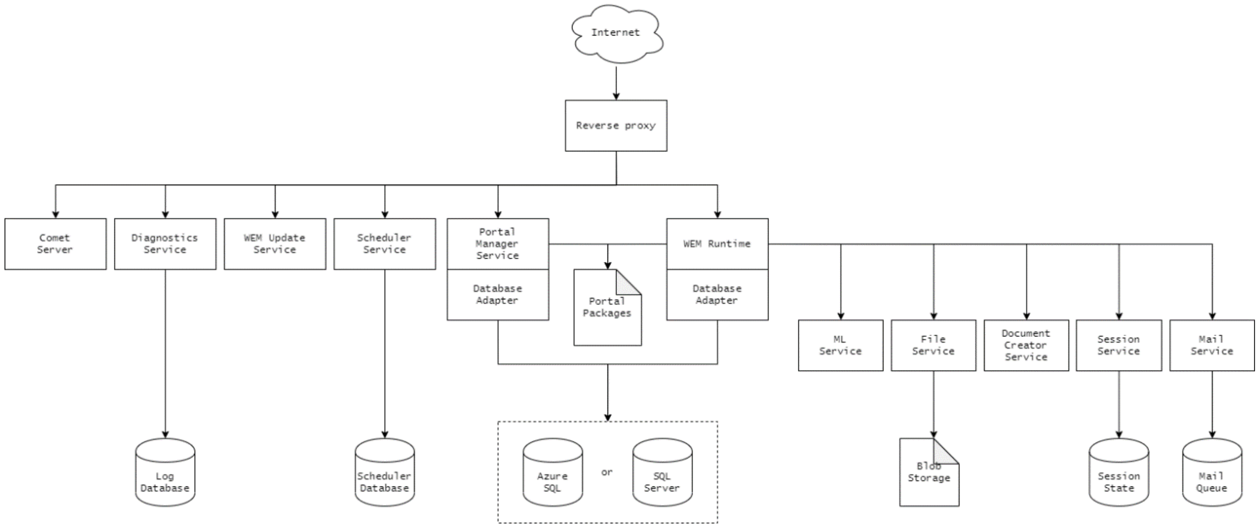
This can be done in 2 ways.

Whenever a user tries to login but fails because of a wrong password a message could be “Incorrect password”. But if a login fails because the account is unknown the message could be “Account not found”. This means a hacker can test which accounts are present in the system. This is sometimes mentioned as a vulnerability. Best practice is therefore to not give away any clues about the account being present or not. A message could therefore be “Username password combination not found”.

Another issue could happen at forgot password. Whenever a user clicks on forgot password a screen is usually shown asking for the e-mail. When the user request that new password a message could be shown like “An e-mail has been sent to your account” and sometimes when the user is unknown a message could be “No account found”. Here it is again easy to mitigate by changing the message into something like “If this account is known an e-mail has been sent to the corresponding e-mail address”

5 Runtime infrastructure

The following diagram provides a high-level overview of a WEM runtime environment:



The reverse proxy runs behind a firewall In a separate subnet (DMZ). There are several components that expose public endpoints through the reverse proxy server:

- WEM runtime – runs the actual application/s
- Comet server – message broker for real-time messages
- Diagnostics service – a secure API that exposes logging information
- WEM update service – a secure API that is used to distribute WEM runtime updates
- Scheduler service – a secure API to schedule cronjobs (that are predefined as part of the WEM application)
- Portal manager service – a secure API that is used to publish application updates

The WEM runtime components run in a separate subnet. Only the components that expose a public endpoint are accessible by the reverse proxy.

All components are stateless. State is delegated to external databases (SQL), in-memory data storage (redis) or blob storage. This makes it easy for components to scale and run in a load-balanced cluster or fail-over cluster.

5.1 Network

Our shared runtime environment contains three subnets:

- DMZ – contains reverse proxy. This is the only subnet with ingress network traffic from the internet (behind a firewall). The ingress traffic is restricted to port 80 and 443. The DMZ has restricted access to the application subnet.
- Application subnet – contains all the WEM application services.
- Data subnet – contains the external databases. This subnet is only accessible from the application subnet.

5.2 Service accounts

All WEM components run under a specific service account that has restricted user rights that are tailored for the specific needs of the component.

5.3 Backups

Database backups are stored in a separate data center and tested regularly. All databases are backed up using the "full recovery model". This allows point-in-time recovery of a database.

- Full backups are performed on a regular basis
- Incremental backups are performed on a regular basis
- Transaction log backups are performed on a regular basis

5.4 Access

Direct access to our shared production runtime environment is only possible through a VPN connection from a single VM in our development network. Access to the production runtime environment is restricted to three WEM employees.

6 Physical

Besides the technical security, several physical measures are in place to ensure our (and your) data is secure.

6.1 Storage of data

Just like other browser-based web application there is no storage of data on the local device beyond what is currently displayed on the screen. WEM Project models and data are always only stored on the WEM database cluster you are currently working with and only accessible through your WEM project.

6.2 Physical access

Access to the WEM building and the office floors is restricted. People only have purpose assigned digital keys which allows access to approved offices. The building is secured with a security company which monitors the building with cameras as well as with physical security guards.

6.2.1 Host buildings

Access to buildings where data and logic is stored is restricted. Access is only allowed by appointment and only for people who are known.

WEM and its distributors, partners and clients do not have access to the physical locations hosting the applications.

7 People

Regarding Safety & Security, people are our most critical and precious factor.

7.1 Background check

Every WEM employee is subjugated to a background check which consists of reaching at least two referents. A statement of conduct of good behavior issued by the Dutch authorities is required in order to work at ZoomBIM/WEM for government projects.

7.2 Employment Agreement

Every employer's contract includes standard paragraphs on responsibilities on secrecy, security, and intellectual property.

7.3 Employee Manual

A key part of the employment agreement is the Employee Manual which documents expected behavior and requirements in key situations. This includes subjects including:

- Data leakage procedure
- Intellectual property and Secrecy
- How to act on Data Security
- Behavioral code on social media

These topics will still be effective, even when leaving the organization.

There are also measures in place to help people make the right decisions when tempted by external influences.

8 Risk Management

8.1 Proactive vulnerability assessment

On a regular basis, but at least every 6 months, a proactive assessment on our safety measures is conducted. All parties that are involved around Safety & Security will gather and review all aspects of our logical, physical, and procedural safety, security and data.

As part of this review new developments in state of the art as well as possible new threats are reviewed and assessed for impact and applicability.

Results from this review are commonly further research or active improvement efforts carried out by individual team members with results reported back to the full group.

All our employees (involved in Safety & Security) actively keep their knowledge on applicable topics up-to-date and are given the time and budget to do so as part of their regular work duties.

8.2 External Factors

Security is an ever-evolving area and new information, and threats can come to our attention at any time. All periodic review described in this document and all existing measures to ensure security, safety and privacy of data can and will be updated at any time if and when we received new or updated information that challenged our current solutions. This is a reactive policy as you cannot plan for unknown events.

8.2.1 DDOS

A DDOS attack is a typical event that cannot be predicted but we can anticipate on. There are mechanisms that quickly recognize an attack and take counter measures immediately, without human interaction.

DDOS and similar attacks are a constant in the normal day-to-day operation of the WEM platform, we have taken numerous steps to ensure that the impact of these external factors is minimized, including:

- Multiple network paths

- Active and reactive filtering early in the process
- Data abstraction layers to easily recognize malformed request
- Over provisioned capacity
- And others.

We will not discuss the details of these security measures as those would provide potential attackers with potential knowledge to perform a precise disruption attempt.

8.2.2 Viruses and mall-ware

Creators of viruses are always ahead of the game for one cannot react on something that isn't there yet. Therefor it is important to select an antivirus company with the shortest reaction time and that is exactly what WEM did.

Because of the single function setups of all the servers in the WEM environment the likelihood of virus or malware infection is small, we have however taken the necessary precautions to limit the risk and limit the impact if such an event may occur.

8.3 Disaster Recovery

Disaster can always strike at a moment one least expects. Such an event is impossible to predict and so we don't try. What we did do however is setup a system that takes over should our entire infrastructure fail.

The key infrastructure is hosted in at least 2 active sites at any one time, the DR infrastructure is hosted with another provider and completely separated, logically, legally and physically.

8.4 WEM Waarborg

As a failsafe, WEM uses a totally different environment that contains up-to-date backups of the WEM applications and databases, WEM source code and WEM binaries.

This environment is managed by an independent trust, WEM Waarborgfonds, that provides continuity of services in case of operational, legal or financial issues with the ZoomBIM holding group of companies.

All customers that have signed on to the no-cost standard SLA are covered as part of the DR and WEM Waarborg services.