Doc. name :	**Oxygis security**
Version :	18/06/2025
Summary :	Your security is very important to us! Here is a summary of what we do every day to ensure that your data is safe with Oxygis and that we apply the best security practices on our hosted version, the Oxygis Cloud.

1 - Oxygis Cloud (Oxygis Cloud Infrastructure)

I.Backups / disaster recovery

- Weekly back-ups stored for 1 year
- Backups are stored on Microsoft Azure
- Backups are replicated in at least 3 different data centres in an availability zone in Western Europe. On request, automatic replicas can be made in other availability zones (this may give rise to additional costs, which will be passed on to the Customer).
- You can contact our helpdesk to restore any of these backups to your active database (or to the side).
- The actual locations of our data centres are specified in our ***Privacy Policy***.
- **Hardware failover:** no services are hosted directly on baremetal, but only on virtual servers or managed services from Microsoft Azure or Amazon Web Services.
- **Disaster recovery: in** the event of a complete disaster, with a data centre *completely out of service for an extended period,* preventing failover to our local hot-standby (this has never happened so far, it's the worst-case scenario), we have the following objectives:
  - **RPO** (Recovery Point Objective) = 24 hours. This means that you can lose a maximum of 24 hours' work if the data cannot be recovered and we have to restore your last daily backup.
  - **RTO** (Recovery Time Objective) = 24 hours for paid subscriptions, 48 hours for free trials, educational offers, freemium users, etc. This is the time required to restore service in another data centre if a disaster occurs and one data centre is completely out of service.
  - How it's done: we actively monitor our daily backups, and they are replicated to multiple data centres within an availability zone. We have an automated provisioning system to deploy our services to a new hosting site. The restoration of data based on our previous day's backups can then be carried out in a matter of hours (for the largest clusters).

II.Database security

- Customer data can be stored in a dedicated database - no data sharing between Customers.

III.Password security

- Customer passwords are protected by industry-standard PBKDF2+SHA512 encryption (salted + stretched for thousands of turns).
- Oxygis staff do not have access to your password and cannot retrieve it for you. The only option if you lose it is to reset it.
- Login details are always transmitted securely via HTTPS.
- Password policies: passwords must be at least 10 characters long and contain at least one uppercase letter, one lowercase letter, one number and one special character.

IV.Staff access

- Oxygis support staff can log into your account to access the settings related to your support issue. To do this, they use their own credentials, not your password (which they have no way of knowing).
- This special staff access improves efficiency and security: they can immediately reproduce any problem you encounter, you never have to share your password, and we can audit and monitor staff actions separately!
- Our helpdesk staff do their utmost to respect your privacy and only access the files and parameters needed to diagnose and resolve your problem.

V.Physical security

Oxygis Cloud servers are hosted on Microsoft Azure, no physical access is possible.

VI.Communications

- All data communications to Client instances are protected by state-of-the-art 256-bit SSL (HTTPS) encryption.

2 - Oxygis (the software)

I.Software security

Oxygis is coded in C# for the backend and AngularJS for the frontend.

Oxygis' R&D processes include code review stages that include the security aspects of new code elements as well as existing elements.

II.Safety by design

Oxygis is designed to prevent the introduction of the most common security flaws:

- SQL injections are prevented by using a higher-level API that does not require manual SQL queries.
- XSS attacks are prevented by the use of a high-level modelling system that automatically escapes injected data.

See also the **OWASP Top Vulnerabilities** section to see how Oxygis is designed from the outset to prevent such vulnerabilities.

III.OWASP's top vulnerabilities

Here is Oxygis' position on the main Web application security issues, as listed by the **Open Web Application Security Project** (OWASP):

- **Injection vulnerabilities:** Injection vulnerabilities, particularly SQL injection, are common in Web applications. Injection occurs when user-supplied data is sent to an interpreter as part of a command or query. The attacker's hostile data causes the interpreter to execute unwanted commands or modify data.

  Oxygis uses an ORM that abstracts the construction of queries and prevents SQL injections. Developers do not normally manually construct SQL queries, which are generated by the ORM, and parameters are always correctly escaped.

- **Cross Site Scripting (XSS):** XSS flaws occur when an application takes user-supplied data and sends it to a web browser without first validating or encoding that content. XSS allows attackers to execute scripts in the victim's browser, which can hijack user sessions, deface websites, possibly introduce worms, etc.

  Oxygis escapes all data, which prevents XSS.

- **Cross Site Request Forgery (CSRF):** A CSRF attack forces a connected victim's browser to send a fake HTTP request, including the victim's session cookie and any other automatically included authentication information, to a vulnerable web application. This allows the attacker to force the victim's browser to generate requests that the vulnerable application thinks are legitimate requests from the victim.

  Oxygis includes an integrated CSRF protection mechanism. It prevents any HTTP controller from receiving a request without the corresponding security token. This is the recommended technique for CSRF prevention.

- **Malicious file execution:** Code vulnerable to remote file inclusion (RFI) allows attackers to include hostile code and data, leading to devastating attacks, such as total server compromise.

  Oxygis does not provide functions for remote file inclusion.

- **Insecure direct object reference:** A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, database record or key, in the form of a URL or form parameter. Attackers can manipulate these references to gain unauthorised access to other objects.

  Access control to Oxygis is not implemented at the user interface level, so there is no risk of exposing references to internal objects in URLs. Attackers cannot bypass the access control layer by manipulating these references, as each request must always pass through the data access validation layer.

- **Insecure cryptographic storage:** Web applications rarely use cryptographic functions appropriately to protect data and credentials. Attackers use weakly protected data to commit identity theft and other crimes, such as credit card fraud.

  Oxygis uses a standard secure hash for user passwords (by default PKFDB2 + SHA-512, with key stretching) to protect stored passwords. It is also possible to use external authentication systems to avoid storing user passwords locally.

- **Insecure communications :** Applications often fail to encrypt network traffic when it is necessary to protect sensitive communications.

  Oxygis works with HTTPS.

- **Lack of URL access restriction:** Often, an application protects sensitive functionality only by preventing the display of links or URLs to unauthorised users. Attackers can use this weakness to gain access and perform unauthorised operations by accessing these URLs directly.

  Oxygis access control is not implemented at the user interface level, and security does not rely on hiding special URLs. Attackers cannot bypass the access control layer by reusing or manipulating a URL, as each request must pass through the data access validation layer. In the rare cases where a URL provides unauthenticated access to sensitive data, such as the special URLs that Customers use to confirm an order, these URLs are digitally signed with unique tokens and sent only by email to the intended recipient.

IV.Reporting security vulnerabilities

If you need to report a security vulnerability, please report it to our Help Desk as soon as possible. These reports are treated with the highest priority, the issue is immediately assessed and resolved by the Oxygis security team, in collaboration with the reporter, and then responsibly disclosed to Oxygis Clients and users.

Update and contact

This document is updated regularly to reflect developments in our safety practices. If you have any questions or would like to find out more, please contact us at info@oxygis.eu.

Related documents

- Privacy policy
- Responsible disclosure policy
- Service Level Agreement (SLA)
- Subscription agreement

Additional information on safety

The cloud infrastructure used by Oxygis is based on Microsoft Azure data centres that are ISO 27001 and SOC 2 certified and comply with industry security standards. Access to production environments is strictly limited to authorised members of the technical team, all of whom are subject to an internal confidentiality agreement. Access logs are kept for analysis in the event of an incident.

Safety tests

Oxygis carries out regular security tests, including automated vulnerability scans on its production environments, to ensure minimal exposure to known risks.

Authentication and access management

Oxygis recommends the use of strong passwords and two-factor authentication (2FA) for access to its platforms, when available. User roles enable fine-tuned management of access rights to functions and data.