tinQwise

Version 2025.07

TinQwise Policy

004.3
Operational Resilience & Disaster Recovery

TinQwise B.V. | Mussenstraat 15 | 1223 RB | Hilversum | The Netherlands | +31 35 538 5656
CoC Hilversum 32082603 | VAT NL809771287B01 | IBAN NL26 INGB 0006679651 | SWIFT / BIC INGBNL2A | tinqwise.com

tinQwise

Validity and document management

Version	Date	Author	Description
01	2021-02-21	Simon Kennedy	Document creation
02	2021-10-28	Simon Kennedy & Emil de Valk	Annual review and policy update.
03	2022-11-04	Simon Kennedy, Emil de Valk, Okke Formsma, Elena Neagu	Annual review and policy update
05	2023-03-09	Emil de Valk and Reinoud van Dommelen	Update to align with standard customer contract set. Definitions, renumbering and consistency.
06	2023-11-09	Okke Formsma and Simon Kennedy	Annual review and policy update
07	2023-12-12	Emil de Valk	Annual review

This document is valid as of January 1st, 2025.

The owners of this document is the C.O.O., who must check and, if necessary, update the document at least once a year.

When evaluating the effectiveness and adequacy of this document, the following criteria need to be considered:

- Uptime against SLA standards
- Overall number of (potential) uptime incidents – via alerting and tracking
- Number of uptime incidents arising from failed controls built into the systems
- Customer (IT organisation) feedback

TinQwise reserves the right to update this Policy to reflect changes in our practices and regulatory requirements. Data subjects will be informed of material changes. Previous versions of this procedure will be stored for a period of 5 years, unless specified otherwise by legal or contractual requirement.

tinQwise

Table of Contents

tinQwise

1. Relevant to

Management	Finance / Legal	HR	Office Management	Marketing, Sales and Account Management	Professional services	IT Support	Product & Engineering	DevOps	Support & Maintenance
X						X	X	X	X

2. Purpose

This document defines the approach to Operational Resilience implemented by TinQwise.

3. Scope

Operational resilience is the ability to provide continuous service through people, processes, and technology that are aware of and adaptive to constant change. It is a real-time, execution- oriented norm embedded in the culture of TinQwise that is distinct from traditional approaches in Business Continuity, Disaster Recovery, and Crisis Management which rely primarily on centralized, hierarchical programs focused on documentation development and maintenance.

TinQwise is responsible for ensuring that our SaaS Application is continuously available, as well as ensuring that we are prepared to handle a wide range of events that could potentially affect our application or infrastructure.

4. Maintaining operational resilience & service continuity

The SAAS Application is architected to prevent outages and incidents, and uptime is a critical component in the design of the application architecture. This includes self-healing and auto-scaling technology, as well as making use of key Amazon Web Services (AWS) technology to ensure continuous uptime. This means that if disruptions do occur, their impact on customers and the continuity of services is as low as reasonably practicable.

A key component of this strategy is using the world class services and tools of AWS. To avoid single points of failure, AWS minimizes interconnectedness within their global infrastructure. AWS's global infrastructure is composed of 20 geographic Regions, which are composed of 61 Availability Zones (AZs), which, in turn, are composed of data centers. The AZs, which are physically separated and independent from each other, are also built with highly redundant networking to withstand local disruptions. Compared to traditional on-premises environments today, the locational diversity of AWS's infrastructure greatly reduces geographic concentration risk.

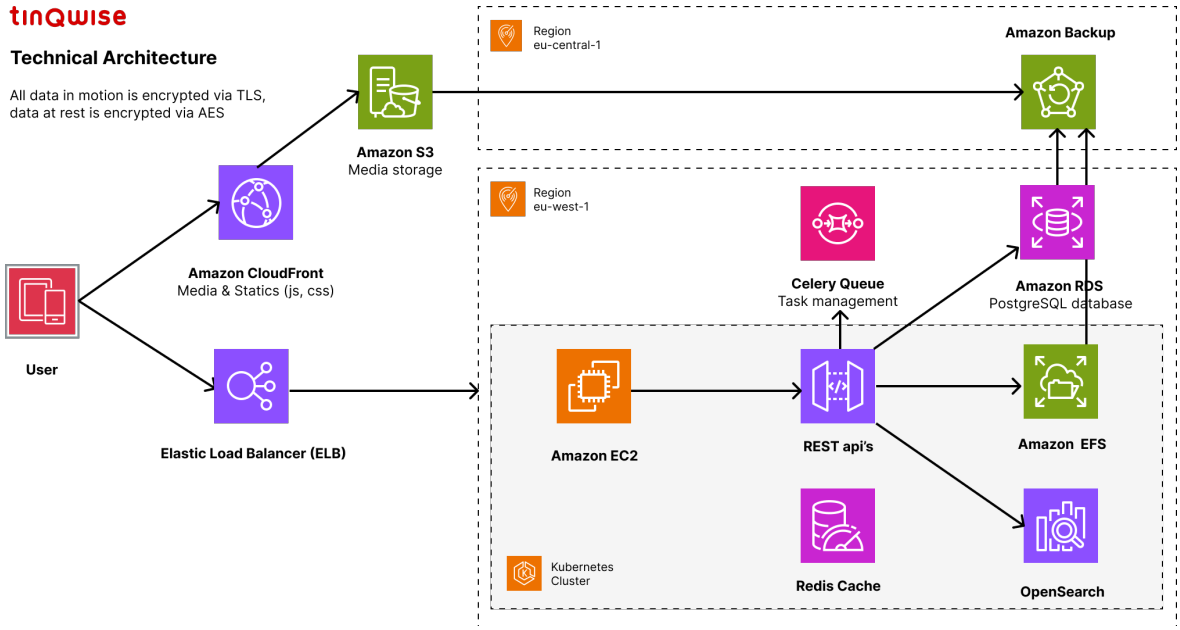
To comply with GDPR ruling and customers data privacy needs, TinQwise only makes use of AWS infrastructure in western Europe for application instances and databases – both production, failover, and backup.

4.1 Application architecture

The SAAS Application is architected to be fully scalable, performant, and resilient using the most modern cloud technology. We use elastic load balancer at the front end, which routes traffic only to healthy and performant instances. Application instances are all run on advanced autoscaling and self-healing infrastructure via Kubernetes and AWS. The cluster is self-healing, with new instances automatically added or removed based on load and health status. As load increases, so additional Kubernetes instances are automatically scaled out and start managing load directed via the Elastic Load Balancer.

The Amazon RDS database scales vertically using AWS autoscaling technology, and caching (Redis) and queuing (Celery) tech are used to ensure optimal performance.

All media and statics are managed independently via the globally scalable CloudFront content delivery network and served directly to the end user.



4.2 Availability incident management

Although the likelihood of such incidents is very low, the TinQwise team is fully prepared to manage events that affect our application uptime. The TinQwise team becomes aware of incidents or degradations in service based on continuous automated monitoring through metrics and alarms via a dedicated suite of monitoring tools – for example AWS CloudWatch at the infrastructure level, and Sentry and New Relic at the application level.

In case of a significant event, an on-call engineer convenes a call with problem resolvers to analyse the event to determine if additional resolvers should be engaged. A call leader drives the group of resolvers to find the approximate root cause to mitigate the event. The relevant resolvers will

perform the necessary actions to address the event. After addressing troubleshooting, repair procedures, and affected components, the call leader will assign follow-up documentation and actions and end the call engagement. The call leader will declare the recovery phase complete after the relevant fix activities have been addressed. The postmortem and deep root cause analysis of the incident will be assigned to the relevant team. Post-mortems are convened after any significant operational issue, regardless of external impact, and root cause analysis and corrective actions are documented and planned for implementation in current or coming sprints. Implementation of the preventative measures is managed and tracked as part of our standard roadmap and sprint planning process.

4.3 Risk scenarios

The primary risks and event scenarios that are considered in the SAAS Application disaster recovery plan include:

- **Natural or other disasters impacting physical infrastructure**: Natural disasters such as floods, storms, or earthquakes could cause significant damage to data centres, servers, and infrastructure, resulting in service disruptions or outages. Example Scenario:
    - Physical damage rendering the AWS Dublin data centre inoperable / offline

- **Cyber Attacks:** Despite the extensive security measures we have in place, Cyber-attacks such as hacking, malware or ransomware could compromise the uptime of our application and data. Example Scenario:
    - Cyber-attack on AWS rendering the AWS DDC (temporarily) inoperable
    - Cyber-attack on TinQwise removing access to AWS production account / removing databases

- **Human Error**: Accidental or intentional human errors such as misconfiguration, data loss, or application downtime could potentially have a significant impact on our application and services. Example Scenario:
    - Accidental production database or infrastructure deletion

- **Hardware or Software Failure:** Hardware or software failures such as server crashes, storage failures, or network outages could cause downtime or service disruptions. Example Scenario:
    - Hardware or software failure rendering the AWS Dublin data centre inoperable / offline

4.4 Recovery time & Recovery point objective (RTO & RPO)

For a majority of TinQwise Customers, a learning experience platform is **non-critical business application,** and our stated RTO and RPO takes this into account.

Recovery Time Objective (RTO)
TinQwise defines RTO is the maximum acceptable downtime for an application after a disaster occurs. Specifically, the time taken to restore the SAAS Application and its data to the point where users can access the application again.

This RTO takes into consideration the potential impact on customer satisfaction, revenue loss, and reputation damage that could occur due to prolonged downtime.

Our RTO is 24 hours.

Recovery Point Objective (RPO)
TinQwise defines RPO is the maximum data loss that an organization can tolerate after a disaster occurs. In the case of the SAAS Application, the RPO can be defined as the amount of data that can be lost after the application is fully restored.

The RPO for SAAS Application is set at **24 hours**. This means that in the event of a disaster, we can tolerate up to 24 hours of data loss. This RPO takes into consideration the potential impact on customer satisfaction, compliance, and business operations that could occur due to significant data loss.

4.5 Redundancy

The entire SAAS Application setup is duplicated in our stand-alone QA environment. The QA environment effectively operates as a redundant set up and is maintained on a separate AWS account and in a geographically separate data centre. The database backup and restore / recovery process is fully tested every two weeks when all production data is restored to a refreshed QA environment for testing prior to bi-weekly releases.

4.6 Backup and recovery approach

Our SaaS Application backup and recovery approach focusses on three key areas:

- Infrastructure & infrastructure configuration “infrastructure as code”
- Databases
- Application code & CI/CD pipeline

Infrastructure:

As described in chapter 4.1, our infrastructure is scalable and resilient to hardware failure on datacentre level, with backup / failover to separate AWS account running in a separate AWS data centre (Frankfurt).

Our Recovery Time Objective (RTO) of 24 hours includes time for completion of a documented set of automated (Infrastructure as code) and manual steps to configure build and test our entire infrastructure set up.

Key steps – Infrastructure:

- Create new AWS account
- Provision key AWS infrastructure elements:
    - Load balancers (+SSL certificates)
    - EC2
    - CloudFront
    - S3 for statics
    - EFS for media

    - RDS for write and read-only database
    - AWS backup
- Provision REDIS (Cache)
- Provision Celery (queuing)
- Provision self-healing Kubernetes cluster
- Deploy application via separate build server and automated CI/CD pipeline

Our live QA environment is hosted on separate AWS account running in a separate AWS data centre. This is a duplicate of our production set up and provides an additional level of redundancy.

Databases:

SAAS Application database setup consists of the following key components:

- Primary application 'write' database
- Read-only database (copy)
- EFS storage for media
- S3 storage for statics
- AWS backup (S3 storage)
- (CloudFront distributed databases for optimal content performance)

All core databases are backed up daily to a separate AWS account running in physically separate AWS availability zones within Europe.

Data retention is 1 week for daily backups. Monthly backups are retained for 6 months.

The entire database setup is duplicated in our stand-alone QA environment. The entire Database backup and restore / recovery process and content is fully tested every two weeks when all production data is restored to a refreshed QA environment for testing prior to bi-weekly releases.

Key steps – Database:

- Restore primary application 'write' database
    - Test & validate restore (automated test script)
    - Read-only auto restores from primary
- Restore EFS database for media
- Restore S3 database for statics
- CloudFront auto populates

Application code & CI/CD pipeline:

The SAAS Application code base, CI/CD pipeline and build server is hosted on stand-alone infrastructure. Backup and alternatives are hosted on local machines and as an additional fail safe, all application code is securely backed up daily at an independent escrow provider.

Key steps – Application code & CI/CD:

- Automated deployment following configuration of new infrastructure and database restore.

tinQwise

4.7 Testing

Database restore and recovery procedures are tested to a production environment bi-weekly on Mondays through the QA refresh process.

Code deployment and CI/CD is tested and executed multiple times per day. Code restore from backup is tested bi-monthly.

The full Infrastructure restore process is tested on a (minimum) 6-monthly basis.

4.8 Incident Response team

Serious incident response is led by the C.O.O. and head of engineering. The team and key responsibilities are as follows:

- **C.O.O.:** End accountability for Serious Incident management
- **Head of engineering:** Overall coordination of the technical response and recovery process
- **DevOps engineers:** Infrastructure, database, code recovery and restoration
- **Customer Service team:** General customer updates in coordination with the marketing and communications team. Updating and maintaining online status page. Response to specific customer requests and tickets per service desk processes.
- **Marketing and communications:** General customer updates and information
- **Account managers:** specific customer engagement, updates and communication.

For more information, please refer to TinQwise' Incident Management Policy.

4.9 Communications

Key communications procedures and pathways will follow TinQwise' standard incident response approach, communication paths and procedures. Additional process steps and communication protocols added appropriately for the severity of the disaster.

Internal communication:

- Disaster notification via standard automated monitoring and alerting process
- Upon confirmation of a full outage by service desk and DevOps team, immediate escalation to C.O.O., Head of Engineering via slack, mobile phone
- Immediate notification via slack and mobile phone to the **incident response team**
- Immediate notification via slack to account managers and all impacted employees / teams
- Immediate slack 'huddle' as operations room for the incident response team
- Periodic updates to account managers, Marketing and communications
- Notification to all TinQwise upon resolution.

Customer communication:

- Upon confirmation of a full outage by service desk and DevOps team, Immediate update of the online status page and automatic notification to customers via the service desk
- After initial investigation/analysis, direct updates to key accounts via dedicated account managers
- Periodic progress updates via online status page and automatic notifications to customers

- Periodic direct updates after key events or progress steps to key accounts via dedicated account managers
- Upon platform restoration:
    - Immediate update of the online status page and automatic notification to customers via the service desk
    - direct updates to key accounts via dedicated account managers
- After completion of investigation and root cause analysis:
    - update of the online status page and automatic notification to customers via the service desk
    - direct updates to key accounts via dedicated account managers

5. Organisation of operational resilience management

5.1 Resilience governance body

The C.O.O. and H.o.E. are ultimately accountable for operational resilience and uptime. Day to day operational responsibilities for uptime and operational resilience are taken care of by the DevOps engineers and TinQwise product team via the product board.

The TinQwise Product Board coordinates activities throughout TinQwise ensuring that appropriate architecture, product development policies and resilience measures are in place to support the company's uptime and resilience needs.

5.2 Related roles and responsibilities

C.O.O.

The C.O.O. ensures uptime and operational resilience through:

- ensuring that product development requirements and infrastructure and application architecture are established and are compatible with the strategic direction of the organization;
- ensuring the integration of scalable application development and infrastructure management into the organization's prioritization and processes;
- ensuring that the resources and appropriate competencies needed for maximum uptime and operational resilience are available in the engineering team;
- communicating the importance of operational uptime;
- promoting continual improvement; and
- supporting other relevant management roles to demonstrate their leadership as it applies to their areas of responsibilities.

Uptime and resilience are also a key responsibility of all technical employees and contractors/consultants, and this is regularly communicated and reinforced.

Product Board

The Product board is responsible for:

- Ensuring that suitable technical, physical and procedural controls are in place in accordance with operational requirements, SLA's and TinQwise Policies.
- Ensuring that high-performance engineering and correct practice is properly applied and used by all employees, contractors, consultants, temporary, and other staff at TinQwise and TinQwise.
- They take measures to ensure that staff
    - are informed of their obligations to write and publish code that does not negatively impact the performance and stability of the application;
    - comply with the Policies and actively support the associated controls;
    - have code reviewed via peer review process and monitored for potential impacts on performance.
- providing the direction, resources, support, and review necessary to ensure that operational resilience and uptime are appropriately safeguarded within their area of responsibility.
- Monitoring and reporting on the status of operational resilience and application uptime within the organization.

Head of Engineering and DevOps engineers

The H.o.E. and DevOps engineers are responsible for:

- Day-to-day operation of the application infrastructure and overall operational resilience.
- Reviewing logs and following up on any exceptions identified.
- Monitoring and logging performance alerts and events.
- Coordinating performance management and assessments.
- Coordinating performance and load testing.
- Following up with system administrators for the identification, testing, distribution, deployment and implementation of technical configurations, and requirements if appropriate.
- Investigating any performance-related incident, event or behaviour.
- Categorizing, reporting, analysing, responding and escalating performance- or application stability-related incidents.

6. Compliance to this policy

The C.O.O. is responsible for verifying compliance to this policy through various methods, including but not limited to, business tool reports, internal and external audits. Any exception to the policy is subject to the Information security exceptions management process.

An employee found to have violated this policy may be subject to disciplinary action, up to and including termination of employment.

tinQwise

7. Managing records kept on the basis of this document

Record name	Storage location	Person responsible for storage	Controls for record protection	Retention time
Incident reports	SharePoint and / or GitLab	C.O.O.	TinQwise staff can access these files	5 years
Pentest and other security test reports	SharePoint	C.O.O.	TinQwise staff can access these files	5 years
Uptime monitoring and performance logs	AWS dashboards	H.o.E.	TinQwise staff can access these files	2 years