klart

# Trust Center

Security, data protection and compliance documentation for the Klart platform.

| | |
|---|---|
| **Version** | v0.5.8 |
| **Generated** | 2026-11-13 |
| **Source** | docs.klart.com/docs/trust |
| **Contact** | security@klart.com |

## CONTENTS


1. Trust Center
2. Security Overview
3. Architecture Overview
4. Security Controls
5. Data Protection & GDPR
6. Sub-processors & Service Providers
7. Banking & Data Providers
8. Security FAQ
9. Compliance Roadmap


---

This document is a point-in-time export of the Klart Trust Center, generated automatically from the same release that is published on the web. The web version is authoritative and may have been updated since this PDF was generated — check the version above against the live site before relying on it.

# Trust Center

*Last updated: 2026-11-03*

Klart is a platform for working with transaction data — compliance and financial intelligence for law firms, financial institutions, and other organisations that analyse financial activity. This Trust Center documents our security posture as it is today, including what is still on our roadmap — we would rather give a straight answer than a polished one.

The web application and the API share the same backend and the same access-control path, so everything described here applies to both unless a page states otherwise.

## At a glance

- **Hosting**: EU — DigitalOcean, Amsterdam
- **GDPR**: the customer is the data controller; Klart is the data processor. Klart is a trading name — the contracting legal entity is Lexabit AB (SE), or its Danish entity Lexabit ApS where the customer contracts with it
- **Certifications**: see the Compliance Roadmap for current status
- **Banking access**: via regulated open-banking providers, supervised by their national financial supervisory authorities
- **Security contact**: security@klart.com (vulnerabilities, security reviews)
- **Compliance contact**: compliance@klart.com (DPA, data subject requests, sub-processors)

## Explore the Trust Center

- **Security Overview** — architecture, access control, and encryption, as they operate today.
- **Architecture Overview** — components, integrations, infrastructure, and key technical dependencies.
- **Security Controls** — the full control register, with an honest status for each control.
- **Data Protection & GDPR** — what data we process, where it lives, and data-subject rights.
- **Sub-processors & Service Providers** — sub-processors, operational service providers, and banking/data partners.
- **Banking & Data Providers** — the open-banking and company-data providers we connect to.
- **Security FAQ** — answers to common questions from IT and security reviewers.
- **Compliance Roadmap** — current certification status and what's planned.

# Security Overview

*Last updated: 2026-11-03*

Klart is a platform for working with transaction data — compliance and financial intelligence for law firms, financial institutions, and other organisations. This page describes the security architecture as it operates today. Where a control is on our roadmap rather than live, it is labelled **Planned** — we do not present planned controls as current.

## Platform architecture

Klart's backend is a Laravel 13 API; the customer portal and administration panel are React single-page applications that consume it over HTTPS. The platform is hosted by DigitalOcean in Amsterdam, keeping infrastructure and data within the EU.

The application database is MySQL, with a separate self-hosted MongoDB instance used only for public business-registry data — company information and registered-role information from national registries — it does not contain the customer's own client, case, or user data. Background work — transaction auditing, bank-account synchronisation, and health monitoring — runs through Laravel's database-backed queue system. Production runs on its own dedicated infrastructure, fully separated from all non-production environments; development and staging are separate deployments, each with its own configuration and data, hosted on shared non-production infrastructure.

## Tenant isolation

Klart runs on a single shared application database — it does not use a separate database per customer. Isolation is enforced in the application layer by the entity-permissions system on every request: each request carries an `X-Scope` header identifying the entity context (tenant, group, project, client, or case); middleware resolves that header and verifies the authenticated actor can access the requested scope through structural relationship chains and explicit visibility assignments. This follows a four-layer model — Authentication, Visibility, Entitlement, and Action — and scope and permission checks are enforced across the v1 API. Generated reports and documents are access-controlled behind the same permission and scope checks.

## Access control

Access to data and actions is governed by a role-based, fine-grained entity-permissions system covering every application entity — tenants, clients, cases, groups, and projects. Every request is evaluated against the authenticated actor's specific role assignments and explicit visibility grants for the requested scope: results are filtered to what the actor can see, and each action is separately permission-gated, rather than access being implied by authentication alone. These checks are enforced across the v1 API.

## Authentication

User passwords are hashed with Argon2id before storage. API and single-page-application sessions are authenticated using Laravel Sanctum, and application-to-application access uses Sanctum-issued API keys and OAuth2 client-credentials tokens. Sign-in via Google and Microsoft is supported using OAuth with PKCE, where enabled for a tenant. Session cookies are set with the `Secure`, `HttpOnly`, and `SameSite=Lax` attributes in all deployed environments. Login attempts are rate-limited (6 per minute per IP) and password-reset requests are rate-limited separately (5 per 15 minutes); a credential is locked for 15 minutes after 5 consecutive failed login attempts.

Two-factor authentication (2FA) is **Planned** and not yet available.

## Encryption

All traffic to Klart is served over HTTPS (TLS, via Let's Encrypt certificates) with HSTS enabled.

Integration credentials — the secrets Klart uses to connect to external providers — are encrypted at rest using AES-256 via the application's encryption layer. Client national identity numbers are also column-encrypted. Beneath that, all data in the production database is encrypted at rest, and its daily backups are encrypted as well. The application-level encryption keeps credentials and identity numbers encrypted even inside the database itself.

## Banking data & PSD2

Klart retrieves banking data through regulated third-party open-banking (AISP) providers, each supervised by its national financial supervisory authority. The currently active provider is **Enable Banking**; the platform has integration support for additional providers (such as Neonomics), which are only presented as active when they actually are. Customers authenticate directly with their bank during the connection flow; Klart never receives or stores bank login credentials. Connected accounts are covered by individual consents, each with its own status and lifecycle; consents expire and must be renewed, and expired or revoked consents are enforced automatically.

Provider credentials Klart uses to authenticate with our banking providers are encrypted at rest; per-connection session data is held server-side.

Access granted to a connected banking application is fail-closed: an app's permitted actions are hard-capped to an explicit allowed set at connection time, and an unrecognised app is denied by default rather than implicitly trusted. Transaction data retrieved from connected accounts is subject to the same entity-permission and scope checks as the rest of the platform.

## Monitoring & availability

Klart publishes a public `/health` endpoint reporting overall platform status, a token-gated endpoint with detailed diagnostics, and a per-service health check for each core service. The public endpoint is monitored externally by an uptime-monitoring service (UptimeRobot) to alert on availability issues. Background and asynchronous work — including transaction auditing, bank-account synchronisation, and health monitoring itself — runs through dedicated, database-backed job queues rather than inline in the request cycle.

## Security contact

Report vulnerabilities or security questions to **security@klart.com**.

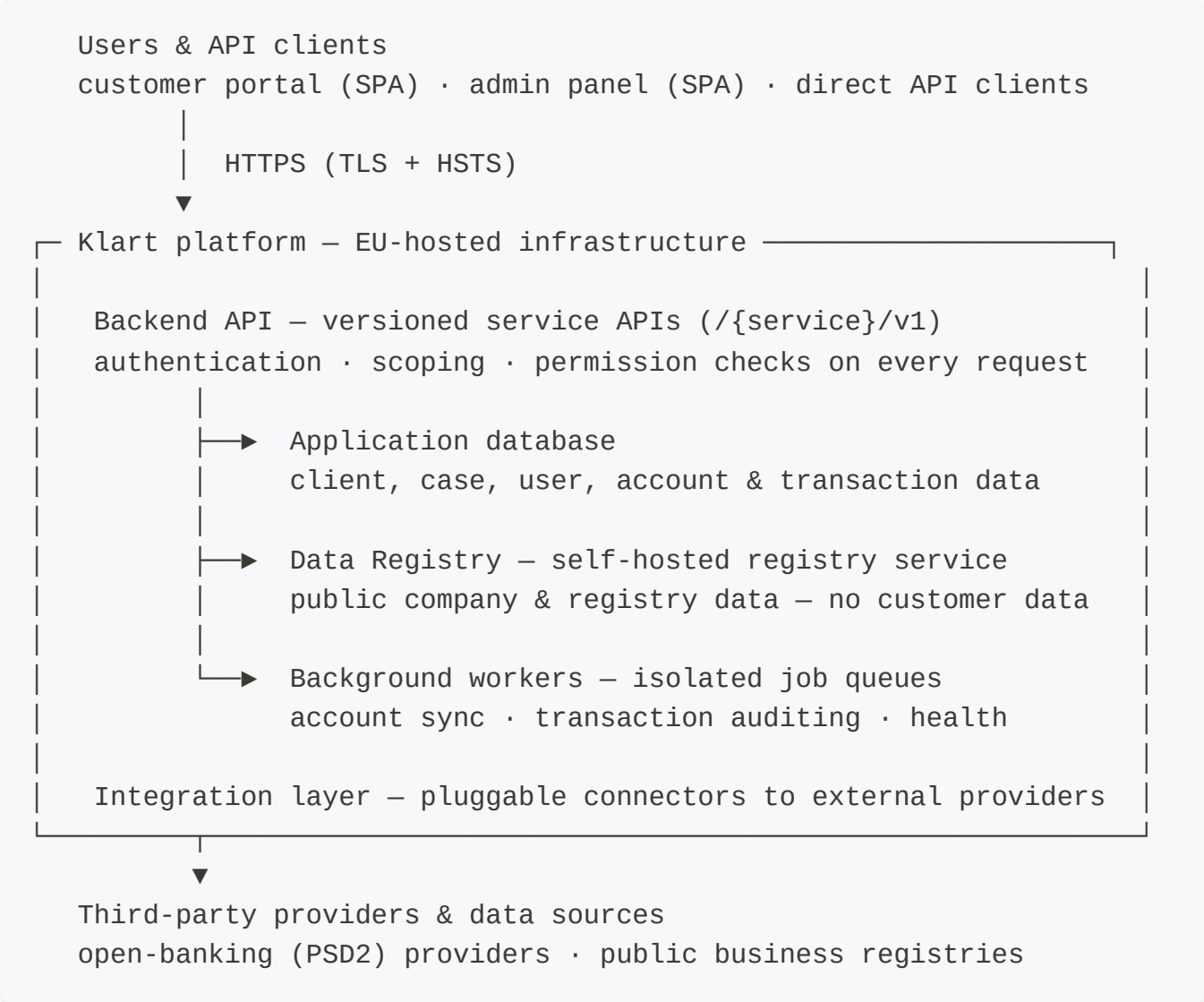
# Architecture Overview

*Last updated: 2026-11-12*

This page describes how the platform actually fits together: the components, how a request travels through them, where data lives, and what the platform depends on. It is written for technical and security reviewers; the security properties of each part are covered in depth in the Security Overview.

## The system at a glance

Everything inside the boundary below runs within Klart's own hosted infrastructure in the EU.




- The **backend** is a single API exposing versioned service APIs (`/{service}/v1/...`). The **customer portal** and **administration panel** are single-page applications with no backend of their own — every capability goes through the same API, so the access-control path below applies to all surfaces equally.

- There is **one shared application database** — isolation between customers is enforced in the application layer on every request, not by separate databases per customer.
- The **Data Registry** is a separate, self-hosted registry service deployed inside the same infrastructure. It is deliberately segregated by content: it holds only public business-registry data (companies, registered role holders, beneficial owners) and never the customer's own client, case, or user data.
- The **integration layer** connects the platform to external providers through a common integration framework — providers are pluggable behind it and engaged per provider. Which providers are active, and on what basis, is documented on Banking & Data Providers rather than hard-wired into the architecture.

## What happens on a request

Every API request passes through the same pipeline before any data is returned:

```
request ──▶ TLS (HTTPS + HSTS)                          transport
encryption   ENC-5
        ──▶ Authentication (session / API token)        who is the
actor?      AUTH-2
        ──▶ Scope resolution (X-Scope header)           which entity
context?
        ──▶ Visibility check                            may the actor
SEE it?
        ──▶ Entitlement & action check                  may the actor DO
this?
        ──▶ Filtered response                           only what the
actor may see
```

The scope, visibility, and entitlement stages are the entity-permissions system's four-layer model (Authentication → Visibility → Entitlement → Action), enforced across the v1 API — including generated reports and documents. The Security Overview describes the mechanism in full.

## Where data lives

| Data | Store | Notes |
|---|---|---|
| Client, case, user, and account/transaction data | Application database (MySQL) | Single shared database; application-layer isolation per request ; encrypted at rest, daily encrypted backups |

| Data | Store | Notes |
|---|---|---|
| Public company & registry data | Data Registry — self-hosted registry service (MongoDB) | Public-registry content only, fed by national business registries; no customer data |
| Profile images (logos and avatars) | DO object storage (AMS3) | The only files stored today. Reports and documents are generated on request behind the same permission checks and are not stored. Signed-URL storage for profile images is **Planned** |
| Bank login credentials | **Nowhere** | End users authenticate at their bank; Klart never receives them |
| Integration/provider credentials | MySQL, encrypted | AES-256 via the credential-management layer |

The full data map, by category and source, is on Data Protection & GDPR.

## Data flows in and out

- **Banking data (in)** — retrieved through regulated open-banking providers under a PSD2 consent authorized in the bank's own flow, by the account holder or another party entitled to access the accounts. Consents have individual lifecycles and expiry is enforced automatically. The currently active provider is listed on Banking & Data Providers.
- **Registry data (in)** — public national business registries feed the self-hosted Data Registry.
- **Email (out)** — transactional notifications via Mailtrap (Railsware).
- **Monitoring (out)** — a public `/health` endpoint is polled externally by UptimeRobot; per-service health checks run internally.

Background work — bank-account synchronisation, transaction auditing, and health monitoring — runs on isolated, database-backed queues rather than inline in the request cycle.

## Environments & deployment

Production runs on its own dedicated infrastructure, fully separated from all non-production environments. Development and staging are separate deployments — each with its own configuration and data — hosted on shared non-production infrastructure. Production releases are tag-based and deployed through a controlled pipeline with environment-scoped secrets and approvals — no ad-hoc production changes.

## Key technical dependencies

| Dependency | Role |
|---|---|
| DigitalOcean (Amsterdam, EU) | Hosting: application server, databases, object storage, CDN |
| Laravel 13 / PHP | Backend framework — parameterized queries, CSRF protection, output escaping by default |
| React | Customer portal and administration panel SPAs |
| MySQL / MongoDB (self-hosted) | Application database / Data Registry storage |
| Laravel Sanctum | API and session authentication |
| Enable Banking | Open-banking (PSD2) account and transaction access |
| Let's Encrypt | TLS certificates; HTTPS with HSTS on all traffic |
| Mailtrap (Railsware) | Outbound transactional email |
| Google Workspace | Internal business email and support correspondence |
| UptimeRobot | External availability monitoring |

Third parties that process customer personal data, with data categories and regions, are listed on the Sub-processors & Service Providers page.

## Questions

Architecture questions from security reviews: **security@klart.com**.

# Security Controls

*Last updated: 2026-11-11*

This page lists Klart's security controls with an honest status for each. **Planned** means the control is on our roadmap, not live today — nothing planned is presented as current. Every status below traces to an internal evidence inventory reviewed against the current codebase and infrastructure.

## Data Security

| Control | Statement | Status |
|---|---|---|
| Encryption in transit | All traffic served over HTTPS (TLS, via Let's Encrypt certificates) with HSTS enabled | In place |
| Credential encryption | Integration credentials encrypted with AES-256 via the application's encryption layer | In place |
| Sensitive-field encryption | Client national identity numbers are column-encrypted | In place |
| Credential key management | Credential encryption supports key versioning and rotation; keys are application-managed | In place |
| Database encryption at rest | All data in the production database is encrypted at rest | In place |
| Backups | Automatic daily database backups, kept for 7 days, encrypted, with point-in-time restore | In place |
| Recovery objectives (RPO/RTO) | RPO 24 hours, RTO 48 hours for the production database | In place |
| Restore testing | Regular, documented restore tests | Planned |

## Infrastructure Security

| Control | Statement | Status |
|---|---|---|
| EU data residency | All hosting in DigitalOcean Amsterdam — droplet, database, object storage, and CDN | In place |
| Environment separation | Production on dedicated infrastructure, fully separate from non-production; dev and staging as distinct deployments with their own configuration and data | In place |
| Availability monitoring | Public health endpoint, per-service health checks, and external uptime monitoring | In place |
| Background processing | Sync, transaction auditing, and health monitoring run on isolated, database-backed queues | In place |

## Access Control & Product Security

| Control | Statement | Status |
|---|---|---|
| Tenant isolation | Entity-scoped access control enforced across the v1 API (four-layer model) | In place |
| Fine-grained permissions | Role-based and per-entity permission checks on individual actions | In place |
| Password hashing | Argon2id | In place |
| Account lockout | 5 consecutive failed login attempts trigger a 15-minute lockout | In place |
| Login rate limiting | Per-IP throttling on login and password-reset endpoints | In place |
| Secure session cookies | Secure, HttpOnly, and SameSite attributes set | In place |
| API authentication | Token-based (Laravel Sanctum) for SPA sessions and application API keys | In place |
| OAuth sign-in | Google and Microsoft sign-in, where enabled | In place |

| Control | Statement | Status |
|---|---|---|
| Two-factor authentication | Not yet available | ● Planned |
| Enterprise SSO (SAML/OIDC federation) | Not yet available | ● Planned |
| User deprovisioning | Tenant administrators can deactivate or remove users directly in the product | ● In place |
| Token & key revocation | Personal access tokens, application API keys, and OAuth consents are customer-revocable | ● In place |
| Connected-app constraints | App permission grants hard-capped to a connector-defined allowed set, fail-closed | ● In place |
| Report & document access control | Generated reports and documents are permission- and scope-gated | ● In place |
| Private media storage | Signed-URL storage for uploaded profile media | ● Planned |

## Banking Data & PSD2

| Control | Statement | Status |
|---|---|---|
| Consent-based account access | Bank access via authorized open-banking providers; authorization happens at the bank, by the account holder or an entitled party — Klart never sees online-banking credentials | ● In place |
| Individual consent lifecycle | Individual consents, each with its own status and lifecycle | ● In place |
| Consent expiry & renewal | Consents expire and are renewed through dedicated flows | ● In place |
| Provider credential encryption | Klart's own provider credentials are encrypted at rest | ● In place |
| Scoped banking data | Bank account and transaction data are behind the same entity-permission scoping as the rest of the platform | ● In place |

## Security Operations

| Control | Statement | Status |
|---|---|---|
| Application audit logs | Integration activity and login metadata recorded | In place |
| Controlled deployments | Tag-based production deploys through a pipeline with environment-scoped secrets and approvals | In place |
| Dependency & code security checks | Dependency security audit and static analysis in the development quality gate | In place |
| Privileged-access audit logging | Formal logging of administrative access to production infrastructure | Planned |
| Central security monitoring & alerting | SIEM-style monitoring and automated alerting on suspicious activity | Planned |
| Scheduled vulnerability scanning | Recurring scanning of application, APIs, and infrastructure | Planned |

## Privacy & Assurance

| Control | Statement | Status |
|---|---|---|
| GDPR roles | Customer is data controller; Klart (Lexabit AB / Lexabit ApS per contract) is data processor | In place |
| Data-map transparency | Processed data categories are published, including registry-sourced data | In place |
| Data subject requests | Handled via compliance@klart.com | In place |
| Data Processing Agreement | GDPR Article 28 agreement entered into with each customer; available on request from compliance@klart.com | In place |
| Privacy policy | Not yet published | Planned |
| External penetration test | Not yet conducted | Planned |

| Control | Statement | Status |
| --- | --- | --- |
| Certifications (ISO 27001 direction) | See the Compliance Roadmap for current status and direction | ● Planned |

For narrative detail on how these controls work together, see the Security Overview. For committed direction and target dates as they're set, see the Compliance Roadmap.

# Data Protection & GDPR

*Last updated: 2026-11-11*

## Roles under GDPR

For customer data processed in the platform, the customer is the data controller and Klart is the data processor. **Klart is a trading name**; the contracting legal entity is Lexabit AB (Sweden) or its Danish entity Lexabit ApS, depending on which entity the customer's agreement is with. Contracts, DPAs and invoices therefore name a Lexabit entity, and the processing described in this Trust Center is the same in either case. Klart enters into a Data Processing Agreement (GDPR Article 28) with each customer. The agreement is available on request from compliance@klart.com.

## Data we process

| Category | Examples | Source |
|---|---|---|
| Account & user data | Login credentials, session and authentication metadata | Created at sign-up / sign-in |
| Client & case records | Client and case records the customer creates in the platform | Entered by the customer |
| Bank account & transaction data | Account and transaction data from connected bank accounts | Retrieved under an authorized PSD2 consent through regulated open-banking providers (currently Enable Banking) |
| Company & registry data | Public company information plus registered role holders (directors, board members — name, date of birth, address) and beneficial owners | Public national business registries via Klart's self-hosted data registry |
| Audit logs | Integration activity and login metadata | Recorded by the platform's logging infrastructure |

## Data residency

Klart is hosted entirely on DigitalOcean in Amsterdam — the application server, database, and CDN all run in the Amsterdam region, with object storage in the DO Spaces AMS3 region. All customer data therefore resides within the EU (Netherlands). The self-hosted MongoDB registry

— on the same Amsterdam infrastructure — holds this public-registry dataset; it does not contain the customer's own client, case, or user data.

## Retention & deletion

Data is retained for as long as the customer's account remains active and used to deliver the platform's features. Logs and audit records are kept for fixed periods, after which they are deleted automatically:

| Record | Retention |
|---|---|
| Application logs | 7 days |
| Debugging data (request and job traces) | 7 days |
| Integration HTTP logs | 14 days |
| Security log (credential access, security notification events) | 90 days |
| Platform audit events | 90 days |
| Audit events for access, permission and feature-activation changes | At least 1 year |
| Financial records (subscriptions, charges, payments) | For the life of the customer account |
| Administrative, integration, file-access and credential audit trails | For the life of the customer account |

Requests for data deletion or erasure are handled operationally by contacting the compliance contact below.

## Data-subject rights

Data subjects can access and correct their personal data through the application's normal features (for example, account and profile settings). Requests for data export or erasure beyond what the application supports are handled by contacting **compliance@klart.com**.

# Sub-processors

Klart uses a small number of sub-processors to deliver the platform, including hosting and outbound email , and lists operational service providers where customer personal data may incidentally appear. Open-banking and company-data providers are used under a different model — see Sub-processors & Service Providers and Banking & Data Providers for details.

# Sub-processors & Service Providers

*Last updated: 2026-11-12*

This page lists the third parties involved in delivering Klart, in three categories that reflect how each one actually touches data. We list deliberately inclusively — where there is any realistic path for customer personal data to reach a vendor, we would rather disclose it than argue it away.

## Sub-processors

Sub-processors are third parties that process customer personal data on Klart's behalf, engaged under GDPR Article 28, as part of delivering or supporting the service. Each processes data only to the extent needed for its stated function.

| Provider | Purpose | Data categories | Region | Status |
|---|---|---|---|---|
| DigitalOcean, LLC | Cloud infrastructure hosting (servers, database, object storage, CDN) | All platform data | Amsterdam, Netherlands (EU) | • Active |
| Railsware Products Studio LLC | Mailtrap — outbound transactional email delivery | Email addresses and notification content | EU/US (provider-operated) | • Active |
| Google Cloud EMEA Limited | Google Workspace — internal business email and customer support correspondence (Google Cloud EMEA Limited is Google's EMEA contracting entity for Workspace) | Correspondence with customer staff | EU/US (provider-operated) | • Active |

## Service providers

Operational tools we use to run the company. They are **not** engaged to process customer data, and no customer data is shared with them by design — but day-to-day work (support coordination, issue tracking) means customer personal data may **incidentally** appear in them, for example in a support ticket that quotes a customer email. We list them for exactly that reason.

| Provider | Purpose | Exposure | Region |
|---|---|---|---|
| Slack Technologies Limited (Salesforce) | Internal team communication | Incidental — customer context may be referenced in internal discussion | EU/US (provider-operated) |
| Linear Orbit, Inc. | Issue tracking and development planning | Incidental — tickets may reference customer context | EU/US (provider-operated) |

Slack also appears in the Data Processing Agreement's approved sub-processor list — it is included there for completeness under the same over-inclusive principle as this page; the actual engagement is as described here.

## Banking & data partners

Regulated parties we retrieve data **from**. They act under their own regulatory obligations and the end user's consent — Klart does not share customer data with them or instruct them to process data on Klart's behalf, so they are not sub-processors in the Article 28 sense. For transparency, every open-banking provider we have built or may build an integration for is listed here with its honest current status (mirroring Banking & Data Providers). Only Enable Banking is in use today.

| Provider | Purpose | Basis | Region | Status |
|---|---|---|---|---|
| Enable Banking Oy | Open-banking (PSD2) account and transaction retrieval | Authorized TPP, supervised by its national FSA; access authorized by the end user at their bank | EEA | **Active** |
| Neonomics AS | Open-banking (PSD2) account and transaction retrieval | Authorized TPP; no data flows today | EEA | **Integration built — not active** |
| Tink AB | Open-banking (PSD2) | Authorized TPP; not integrated, no data flows | EEA | **Not yet integrated** |

| Provider | Purpose | Basis | Region | Status |
|---|---|---|---|---|
| Aiia A/S (a Mastercard company) | Open-banking (PSD2) | Authorized TPP; not integrated, no data flows | EEA | ● **Not yet integrated** |

Enable Banking also appears in the Data Processing Agreement's approved sub-processor list — included there for completeness; the engagement model is as described here and on Banking & Data Providers, and Klart does not instruct it to process customer data on our behalf.

Public national business registries feed the self-hosted Data Registry; how each source is engaged is described on Banking & Data Providers.

MongoDB is self-hosted on Klart's own DigitalOcean infrastructure and is not a third-party sub-processor.

Changes to this list are announced to customers in advance. A downloadable export lives on the Downloads page. Questions: **compliance@klart.com**.

# Banking & Data Providers

*Last updated: 2026-11-12*

Klart connects to a small number of external providers to retrieve banking and company data. These providers are not engaged as sub-processors of customer data in the classic sense described on the Sub-processors & Service Providers page — they either act under their own regulatory obligations and the end user's direct consent, or receive no customer personal data at all. This page explains how each is used.

## Open-banking providers

Klart retrieves banking data through regulated third-party open-banking providers (TPPs), each authorized and supervised by its national financial supervisory authority. The currently active provider is **Enable Banking** (Enable Banking Oy), operating under its own regulatory obligations within the EEA. Account and transaction data is retrieved under the end user's explicit consent, given directly in the provider's or bank's own consent flow, not to Klart.

Other open-banking providers we have built or may build an integration for, with their honest current status:

| Provider | Status |
|---|---|
| Enable Banking Oy | ● **Active** |
| Neonomics AS | ● **Integration built — not active** |
| Tink AB | ● **Not yet integrated** |
| Aiia A/S | ● **Not yet integrated** |

A provider is only presented as active when it actually is. Before any additional provider is put into use, it is added to our sub-processor documentation, customers are notified, and this page is updated.

## Company data sources

Company information in Klart is served from our own self-hosted data registry, fed by public national business registries in the countries where our customers operate. The registry includes public-registry information about companies and their registered role holders —

directors and board members (name, date of birth, address) — and beneficial owners. See Data Protection & GDPR for the full data map.

The platform includes integration support for commercial company-data providers; none is currently active, and this page will be updated before any is put into use.

## Data source catalogue

This catalogue lists the data sources we use or may use, with each source's honest current status: **Active** means data actually flows today; **Not yet in use** means no integration exists and no data is retrieved. A source moves to Active here before it is put into use.

### Business registers

| Source | Contributes | Basis | Status |
|---|---|---|---|
| Brønnøysundregistrene (NO) | Companies, registered role holders, beneficial owners | Public national register | • Active |
| CVR / Erhvervsstyrelsen (DK) | Companies, participants/role holders, beneficial owners, financial statements | Public national register | • Active |
| Bolagsverket (SE) | Companies and registered roles | Public national register | • Not yet in use |
| Skat.dk / Danish Tax Agency (SKAT) | VAT registration status | Public lookup | • Not yet in use |
| Statstidende (DK) | Official gazette notices | Public register | • Not yet in use |

### Property, address & geo

| Source | Contributes | Basis | Status |
|---|---|---|---|
| BBR (DK) | Building and dwelling data | Public register | • Not yet in use |
| Tinglysning.dk (DK) | Property registration | Public register | • Not yet in use |

| Source | Contributes | Basis | Status |
|---|---|---|---|
| Address Register (SDFI) / DAWA (DK) | Address data | Public open data | ● Not yet in use |
| DinGeo (DK) | Geo and property context | Public open data | ● Not yet in use |
| Opendata.dk | Open public datasets | Public open data | ● Not yet in use |

## Sanctions, PEP & adverse media

| Source | Contributes | Basis | Status |
|---|---|---|---|
| EU sanctions lists | Sanctions screening | Public official lists | ● Not yet in use |
| OFAC sanctions lists | Sanctions screening | Public official lists | ● Not yet in use |
| PEP lists | Politically-exposed-person screening | Provider will be named here before use | ● Not yet in use |
| Media searches | Adverse-media screening | Source/engine will be named here before use | ● Not yet in use |

## Licences, authorisations & supervisory registers

| Source | Contributes | Basis | Status |
|---|---|---|---|
| Finanstilsynet (DK) | Financial supervisory register | Public register | ● Not yet in use |
| FSR auditors register (DK) | Authorised auditors | Public register | ● Not yet in use |
| Advokatnøglen (DK) | Lawyers directory | Public directory | ● Not yet in use |
| Indsamlingsnævnet (DK) | Fundraising permits | Public register | ● Not yet in use |

| Source | Contributes | Basis | Status |
|---|---|---|---|
| Haulage permits (DK) | Transport licences | Public register | ● Not yet in use |
| Spillemyndigheden (DK) | Gaming licences | Public register | ● Not yet in use |
| Findsmiley.dk + inspection metadata (DK) | Food-inspection results | Public register | ● Not yet in use |

## Commercial query services

Unlike the public registers above, these are commercial third parties that would **receive a query** (an address or company name) when used — so they get their own disclosure with basis and region before any use:

| Source | Contributes | Basis | Status |
|---|---|---|---|
| Google Street View | Visual address context | Commercial service — receives the queried address | ● Not yet in use |
| Google My Business | Business-listing context | Commercial service — receives the queried company name | ● Not yet in use |

A downloadable export of this catalogue is on the Downloads page.

# Security FAQ

*Last updated: 2026-11-13*

## Where is our data hosted?

Klart is hosted by DigitalOcean in Amsterdam, the Netherlands — the application server, database, and CDN all run in that region, keeping data within the EU.

## Is data encrypted in transit and at rest?

All traffic to Klart is served over HTTPS (TLS, via Let's Encrypt certificates) with HSTS enabled. At rest, all data in the production database is encrypted, and so are its backups. On top of that, integration credentials are encrypted using AES-256 via the application's encryption layer, and client national identity numbers are column-encrypted, so those two categories remain encrypted inside the database itself. The keys for this application-level encryption are application-managed on the production environment; the credential-encryption layer supports key versioning and rotation.

## How is our data isolated from other tenants?

Klart runs on a single shared application database — it does not use a separate database per customer. Isolation is enforced in the application layer: every request carries an `X-Scope` header identifying the entity context (tenant, group, project, client, or case), and middleware verifies the authenticated actor can access that scope through structural relationship chains and explicit visibility assignments before any data is returned. See the Security Overview for the full mechanism.

## Do you support Single Sign-On (SSO)?

Sign-in via Google and Microsoft is supported using OAuth with PKCE, where enabled. Enterprise SSO federation (SAML or OIDC) is **Planned** and not yet available.

## Do you support two-factor authentication (2FA)?

Two-factor authentication is **Planned** and not yet available. Today, account security relies on Argon2id password hashing and an automatic lockout after 5 consecutive failed login attempts.

## Who at Klart can access our data?

Klart is a small team; access to customer data in production systems is limited to operations personnel who need it to run and support the platform. Production infrastructure access is key-based and restricted to that same small group, and production deploys run through a

controlled pipeline with environment-scoped secrets and approvals. Formal audit logging of privileged (administrative) access to production infrastructure is **Planned** — it is not yet in place, and we say so rather than imply otherwise.

Klart staff can see a tracked bank account's transactions only when a user who manages that account has turned on **Include in analysis** in the account's settings. Consent is given per user and can be withdrawn at any time; analysis stops once no consent remains, and existing results are kept. Basic account details (account name, number and bank) are visible to Klart operations staff without that consent, so that we can support the bank connection itself.

## How are tokens, credentials, and other secrets handled?

Secrets fall into three groups, handled differently:

- **Customer bank credentials** — never seen or stored by Klart; authentication happens at the bank (see below).
- **Integration and provider credentials** — stored encrypted (AES-256) through a dedicated credential-management layer that supports key versioning and rotation, and logs credential operations to an audit trail.
- **API tokens and keys** — user sessions and API access use Laravel Sanctum tokens; personal access tokens, application API keys, and OAuth consents can each be revoked by the customer directly in the product, taking effect immediately.

## Can we remove a user and their access ourselves?

Yes. Tenant administrators can deactivate or remove users in the product; a removed or deactivated user loses platform access, and their role assignments no longer grant access to customer data. Personal access tokens and application keys created under the tenant can likewise be revoked by the customer without contacting Klart.

## Is the isolation between customers and cases tested?

Tenant and case isolation is enforced by the entity-permissions system on every request (see how isolation works), and that system is covered by an automated test suite exercising role, permission, visibility, and scope checks, run as part of the development workflow. An external penetration test, which would provide independent validation, is **Planned**.

## Do you have central security logging, monitoring, and alerting?

Partially — and we distinguish honestly between the pieces:

- **Application audit logs** are **In place**: integration activity and login metadata are recorded.

- **Availability monitoring** is **In place**: a public health endpoint plus per-service health checks, monitored externally, with a public status view at status.klart.com.
- **Centralized security monitoring and automated alerting on suspicious activity** (SIEM-style) is **Planned** — not yet implemented.

## Have you defined RPO and RTO?

Yes. For the production database:

- **Recovery point objective (RPO): 24 hours.** Backups are taken daily, so at most one day of data is at risk; point-in-time restore usually narrows this considerably.
- **Recovery time objective (RTO): 48 hours.** This is a deliberately conservative commitment, not a typical restore time.

Regular, documented restore testing is **Planned**.

## How do you handle vulnerabilities, patching, and security testing?

Today: dependency security auditing and static analysis run as part of the development quality gate before changes ship, and framework/dependency updates are applied through the normal development cycle. Scheduled vulnerability scanning of the running application and infrastructure, formalized patch-management cadence, and an external penetration test are **Planned** and tracked on the Compliance Roadmap.

## How do you handle bank credentials?

Klart connects to bank accounts through regulated open-banking providers — currently Enable Banking. Customers authenticate directly with their bank during the connection flow — Klart never receives or stores bank login credentials. Bank connections are covered by individual consents, each with its own lifecycle, and expired or revoked consents are enforced automatically. The credentials Klart uses to authenticate with our banking providers are encrypted at rest.

## Are you ISO 27001 or SOC 2 certified?

Not yet. Klart does not hold ISO 27001 or SOC 2 certification directly; our infrastructure provider, DigitalOcean, holds an independent SOC 2 Type II report, and the data centre facilities it uses are certified to ISO/IEC 27001. See the Compliance Roadmap for current status and direction.

## Have you had a third-party penetration test?

Not yet — an external penetration test is **Planned**. See the Compliance Roadmap for details.

## How are backups handled?

The production database is backed up automatically once a day. Backups are kept for seven days and are encrypted, and the database can be restored to any point within those seven days. Recovery objectives are under Have you defined RPO and RTO?; regular, documented restore testing is **Planned**.

## What is your incident-response process?

Klart publishes a public `/health` endpoint and a per-service health check for each core service, monitored externally by an uptime-monitoring service to alert on availability issues. As a data processor, Klart notifies affected customers of a personal data breach without undue delay and in any event within 24 hours of becoming aware of it. We aim to notify as quickly as possible, well within that limit.

## How do you develop software securely?

Production runs on its own dedicated infrastructure, fully separated from the non-production environments. Development and staging are separate deployments — each with its own configuration and data — on shared non-production infrastructure. The underlying framework (Laravel) provides parameterized database queries, CSRF protection, and output escaping by default.

## Can we get our data out?

Audit-log export for integration activity is available in-product. The platform also generates on-demand reports and documents, which are access-controlled behind the same permission and scope checks as the rest of the platform. For other data export or erasure requests, contact **compliance@klart.com**.

## How do we report a vulnerability?

Email **security@klart.com**.

# Compliance Roadmap

*Last updated: 2026-11-11*

Klart is an early-stage platform. We believe security reviews deserve straight answers, so this page states exactly where we are.

## Today

- No third-party certifications (ISO 27001, SOC 2) yet.
- No completed external penetration test yet.
- Security controls that ARE live are documented in the Security Overview and labelled **In place**.

## Inherited assurances

- **DigitalOcean** (infrastructure): holds an independent SOC 2 Type II report, and the data centre facilities it uses, including Amsterdam (AMS3), are certified to ISO/IEC 27001. DigitalOcean offers a GDPR Data Processing Agreement. See DigitalOcean's trust page for their current certifications.
- **Enable Banking** (open banking): an authorized open-banking provider, supervised by its national financial supervisory authority.

## Planned

- External penetration test.
- A published privacy policy.
- Two-factor authentication (2FA).
- Private, signed-URL document storage for files currently served by unguessable public URL.
- Scheduled vulnerability scanning of the running application and infrastructure, and a formalized patch-management cadence. (Dependency security auditing and static analysis already run in the development quality gate.)
- Central security monitoring and alerting for suspicious activity (SIEM-style).
- Audit logging of privileged administrative access to production infrastructure.
- Regular, documented restore testing of database backups.
- Movement toward ISO 27001 as our customer base grows.

Planned items describe direction only — they are **not part of any current subscription or price**. None have committed dates yet; this page will be updated as that changes.

# Questions

Contact **compliance@klart.com**.