Acra

Data leakage prevention for your apps with centralised data storage.

COSSACK LABS

Essential benefits for:

- **Your clients:** secure storage and processing of their data.
- **Your developers:** security tooling that is simple to operate and convenient to integrate.
- **Your product:** 20 additional lines of code that make it 200 times more secure.

Key features:

- Selective, **searchable encryption** with easy API or transparent proxy integration.
- Decryption as **database proxy service.**
- **SQL firewall** that filters requests and detects attacks.
- **Intrusion/leakage detection** via poison records (honey tokens).

Compatibility:

SQL databases: MySQL, PostgreSQL, Google Cloud SQL, Amazon RDS.

Object stores: filesystems, KV databases, Amazon S3, Google Cloud DataStore.

Languages/frameworks: Ruby, Python, PHP, Go, Java, NodeJS, Objective-C/Swift.

Server OS: CentOS, Debian, Ubuntu.
Application OS: Linux (x86/ARM, Android, iOS).

Overview

Encryption of sensitive and personal data is mandated by regulations (GDPR, HIPAA, PCI DSS) and industry best practices. However, building cryptography into a modern application is often a tedious task, which has a limited security impact and plenty of architectural trade-offs. Acra is here to change it.

Protecting the data lifecycle is a cornerstone of security in any application. Acra was built to mitigate data leakage risks while providing **convenient security** across the whole data lifespan within the application. Acra is **easy to integrate**, requires no significant modifications in the existing code, and provides **reliable encryption** and leakage prevention in your apps.

Typical use cases

- Personally-identifiable information (PII) protection under GDPR demands (articles 25, 32, 33, and 34).
- Secure data aggregation and centralised access control for distributed, microservice-rich applications, multi-application systems, industrial sensors, and other highly heterogenous high-throughput systems.
- General sensitive data protection for web and mobile apps, including financial, medical, e-commerce industries.
- Customisable data security toolkit for distributed applications.

Form-factor and licensing

- **Acra Open-source:** Apache 2 licensed open-core version.
- **Acra Pro:** improved performance, easier scaling, management tools, dedicated support.
- **Acra Enterprise:** improved maintenance, high availability, signed audit logging, pluggable crypto engines, search over encrypted data, SIEM support.
- **Acra Enterprise option kits:** custom extensions for particular use cases: SCADA, TimeSeries/Monitoring, deep mobile integration.
- **Acra-as-a-Service:** managed Acra with secure SQL backend of your choice, pre-configured and integrated.

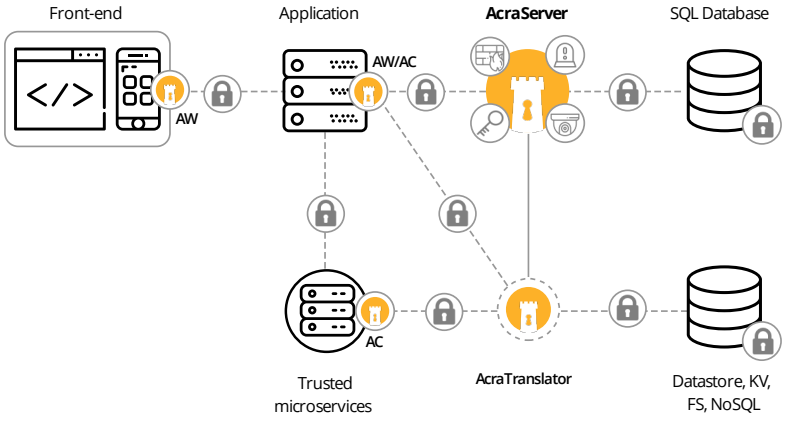
Contact sales@cossacklabs.com for a demo and a quote.

Acra

Data leakage prevention for your apps with centralised data storage.

COSSACK LABS

Architecture and features



Acra easily integrates into modern applications of any complexity, protecting every step of the sensitive data lifecycle in your app:

- **AcraWriter mode:** integrate SDK into client application for secure in-app generation of protected records.
- **Transparent proxy mode:** configure AcraServer to encrypt records in specific columns only.

Core Acra components:

AcraWriter (AW): in-app integration library that encrypts data.
AcraConnector (AC): secure connector that protects transport and provides API interface to AcraServer.
AcraServer: main database proxy that provides decryption, transparent encryption, SQL firewall, and intrusion detection.
AcraTranslator: companion service for decryption of data from non-SQL datastores.

Acra protects only data that needs protection, which is specified within your app code or AcraServer configuration. Call Acra's encryption on the records you need, request them back through Acra — and receive your data safely. Acra decrypts everything in a compartmented server/VM/container, making the keys to the encrypted assets unavailable to an attacker.

Try Acra now!

Acra Live Demo — interactive simulator that demonstrates the basic workflow of Acra.
Visit Acra Open-Source GitHub to see code and examples.

Encryption & key model

Acra integration library encrypts the data in such a way that only the server-side components of Acra can decrypt it. Inside Acra contains all the **necessary key management tools** to support the process: key distribution, rotation, compartmentalisation.

Realistic security model

Acra assumes that datastore and application components can be compromised by attackers, yet the data is protected. Acra **minimises** the leakage scope, **detects** unauthorised behavior, and **prevents** the leakage, informing operators of the incident underway.

Narrow attack surface

Acra's encryption model is built around the requirement that no credentials stored in application components are sufficient for decryption of the data stored in the backends.

As Acra becomes the only gateway for plaintext sensitive data, it can perform various checks to detect anomalous / unauthorised behavior, and log access events to sensitive data.