TECHNICAL OVERVIEW · 01

SEPTEMBER 2026

ACRA Enclaves

A technical overview for regulated financial institutions.

AI extends the enterprise control landscape. Established controls govern identities, devices, networks, applications and stored data. Increasingly capable models and agents add dynamic execution: software that can select tools, traverse systems, process sensitive context and initiate actions.

THE QUESTION HAS CHANGED

BEFORE

~~Who can access a system.~~

NOW, AS WELL

What a system is permitted to reach and do once it is running.

ACRA addresses that question through **enclaves**: controlled execution boundaries for applications, models, agents and data.

IN BRIEF · FOR ALL READERS

01 · DEFINITION

What an enclave is

A hardened, Kubernetes-native boundary that brings identity, network, workload, credential and data controls together around each workload.

02 · MECHANISM

How it works

Applications declare what they need. The platform renders an approved configuration around them and enforces it. Anything more is an explicit, reviewable change.

03 · OUTCOME

What it delivers

Reduced blast radius, separated use cases, less configuration variance and clearer assurance — while each workload keeps full capability inside its approved scope.

04 · RATIONALE

Why it is the right primitive

Enforcement exists before the workload acts and operates outside the model. Behaviour can stay dynamic; authority stays defined.

HOW TO READ · Main text is written for all readers. Panels marked TECHNICAL DETAIL are for architecture, security and platform teams.

§ 01 · DEFINITION

What we mean by an enclave.

An ACRA enclave is a hardened combination of Kubernetes resources and platform controls that defines the environment in which a workload operates. It brings multiple control layers together in one joined boundary spanning:

Ingress and authentication	Kubernetes admission webhooks & runtime configuration
Workload and user identity context	Approved software images
Service-mesh authorisation and audit	Secrets and credentials
Zero-trust authorisation between services	Databases, object storage and backups
Internal and external network paths	Controlled changes to the boundary itself

The operating model is deliberately opinionated:

01 · DECLARE
The application declares the routes, services, images, credentials, storage and external destinations it requires.

02 · RENDER
The platform renders an approved infrastructure configuration around it.

03 · ENFORCE
Admission, identity, network and storage controls enforce that configuration.

04 · EXTEND
Additional requirements arrive through explicit, narrow and reviewable policy changes.

Security posture is therefore established consistently by the environment, complementing application-level controls with a bounded platform foundation.

TECHNICAL DETAIL · HOW THE BOUNDARY IS COMPOSED

An enclave is identified and governed through labelled Kubernetes resources rather than through a single proprietary runtime object. Network policies, ingress routes, middleware, workload policy, service accounts, image rules, secret references and storage resources collectively form the boundary. This allows the control model to remain Kubernetes-native while ACRA supplies the opinionated composition and governance layer.

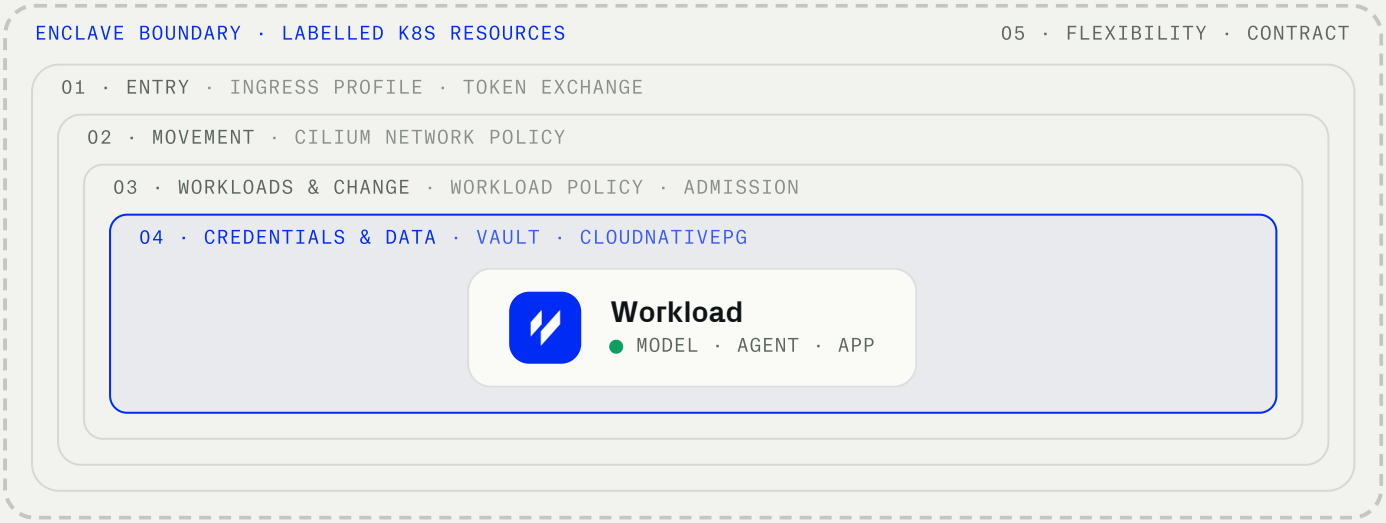

FIG. 1 — The enclave coordinates identity, connectivity, workload policy, credentials and data paths around the execution.

§ 02 · COMPONENTS

The working parts.

01
Controlled entry

Private traffic enters through approved ingress profiles rather than reaching workloads directly. The path combines platform-owned security headers, authentication, token exchange, enclave identity injection and membership checks. Token exchange converts browser or session identity into enclave-scoped context, giving downstream services user, membership, enclave and audience information.

The ACRA service mesh then enforces authorisation and auditing at the service boundary, transparently to the application itself. This gives applications a consistent policy and audit layer without requiring every application team—or every third-party workload—to implement the same control logic independently.

TECHNICAL DETAIL · IDENTITY PROPAGATION

The private HTTP route profile applies an ordered middleware chain before the application handler is called. Token exchange converts the upstream session credential into a short-lived, enclave-scoped token designed to carry issuer, subject, audience, membership and enclave context. The ACRA service mesh validates and applies that context at the service boundary, enforcing authorisation decisions and generating audit records independently of application implementation. The application receives an already-governed request together with the identity context required for its own business logic.

02
Controlled movement

Managed enclaves use Cilium network policy for isolation. Policy defines permitted same-enclave communication, trusted ingress, required platform services, DNS needs and declared external destinations. The managed profile enforces separation between enclave service and data-store paths, making intended connectivity explicit and governable.

Inside the enclave, ACRA extends the same identity-led control model used at ingress to communication between services. Enclave-scoped user credentials are propagated through the system, and the service mesh applies authorisation policy at each service boundary. Services therefore make access decisions from verified user and enclave context rather than from network location alone, with the corresponding interaction captured for audit.

TECHNICAL DETAIL · NETWORK POLICY MODEL

The managed profile combines namespace-level policy with explicit workload egress values. It permits the platform paths required for operation—such as trusted ingress, DNS, Kubernetes services and database management—while maintaining separation between managed enclaves. FQDN-aware policy supports named external services without turning broad internet access into an application default.

TECHNICAL DETAIL · ZERO-TRUST SERVICE NETWORKING

The ingress identity established through token exchange continues across east–west service calls. User and enclave credentials are passed with the request, while service-mesh policy evaluates the calling identity, intended service, audience and permitted operation at each hop. This creates a consistent authorisation and audit model from the initial ingress request through the internal service path.

03

Controlled workloads and change

Admission webhooks evaluate application and infrastructure resources when they are created or updated, before Kubernetes accepts the change. They prevent applications from being applied or edited into insecure configurations and protect the resources that form the enclave boundary, including key namespace labels, ingress resources, network policies and chart-owned security controls. Pod execution is blocked in sealed namespaces, maintaining administration through governed operational paths.

First-party workloads are rendered with hardened defaults where applicable. Workloads must also use approved image paths or explicitly declared third-party image patterns.

- Dedicated service accounts
- Non-root execution
- Read-only root filesystem
- No privilege escalation
- Reduced Linux capabilities
- RuntimeDefault seccomp
- Resource constraints

TECHNICAL DETAIL · ADMISSION WEBHOOKS AND RUNTIME POSTURE

Admission webhooks evaluate Kubernetes create and update requests before the requested configuration is persisted. Policy can prevent changes that remove enclave identity labels, weaken an approved ingress or network path, introduce an unapproved image, or apply a workload outside the required runtime posture. Protected objects include route and middleware resources, Cilium policies, services and workload controllers on the controlled path. Runtime defaults reduce process privilege and filesystem mutability, while dedicated service accounts keep workload identity distinct.

04

Controlled credentials and data paths

Runtime credential use is primarily Vault-centred, while Kubernetes Secrets remain for defined infrastructure integrations. Data services reinforce the same boundary. The current platform includes namespace-local PostgreSQL through CloudNativePG, with a high-availability path and managed failover; object-store-backed backups; namespace-local Redis where enabled; scoped object-storage credentials; explicit bucket ownership; and policy-restricted cross-enclave stateful access.

TECHNICAL DETAIL · CREDENTIALS, STATE AND RECOVERY

Vault roles and workload identity provide the runtime authentication path for application credentials, while infrastructure secrets support defined integrations such as TLS, database access, object storage and registry pulls. CloudNativePG provides the PostgreSQL control plane for replication, failover and backup coordination. Storage credentials and ownership remain scoped to the enclave data path.

05

Controlled flexibility

Required routes, external APIs, images, secrets and storage paths are declared, reviewed and rendered through the platform. This preserves flexibility while making material changes to the boundary visible and governable.

TECHNICAL DETAIL · POLICY AS A WORKLOAD CONTRACT

A declared requirement becomes versioned infrastructure configuration: an ingress profile, network-policy value, approved image pattern, Vault or secret reference, or storage binding. This keeps application intent and platform enforcement connected. A change to the workload's authority is therefore represented as a reviewable change to its contract.

APPLICATION DECLARES	ACRA RENDERS	ENFORCED AT
A private route	→ **Ingress profile · middleware chain**	→ ENTRY · TOKEN EXCHANGE
An external API	→ **FQDN-aware egress value**	→ NETWORK · CILIUM
A container image	→ **Approved image pattern**	→ ADMISSION · WORKLOAD POLICY
A credential	→ **Vault role or secret reference**	→ IDENTITY · VAULT
A database or bucket	→ **Scoped storage binding**	→ DATA · SCOPED CREDENTIALS

FIG. 2 — Applications declare intent; ACRA translates that intent into an approved and repeatable infrastructure shape.

§ 03 · OUTCOMES

What this delivers.

CAPABILITIES	
Compartmentalised execution	Workloads and data paths can be separated by business function, model, use case, project, customer or sensitivity level.
Explicit connectivity	Ingress, egress and cross-enclave paths are governed as declared infrastructure rather than hidden application assumptions.
Zero-trust service networking	Enclave-scoped user credentials and authorisation policies continue across service-to-service calls, creating a consistent control path from ingress through internal services.
Consistent workload posture	Common controls apply across first-party, third-party and AI-generated applications without each team reconstructing the same Kubernetes security decisions.
Configuration integrity	Admission webhooks prevent applications and boundary resources from being created or edited into configurations that fall outside the approved enclave posture.
Scoped identity and credentials	Requests and workloads operate with enclave-relevant identity and authorisation context.
Application-transparent enforcement	The ACRA service mesh applies consistent authorisation and audit controls at service boundaries without requiring those controls to be rebuilt inside each application.
Reviewable change	Boundary-defining resources and additional requirements follow controlled policy paths.
Auditable context	Activity can be associated with user, workload and enclave context.
Workload containment	Separation between enclaves and explicit external connectivity help contain each workload's operating scope.

TECHNICAL DETAIL · EVIDENCE ACROSS LAYERS

The architecture creates complementary evidence at several points: identity and membership at ingress, policy decisions at the Kubernetes and network layers, workload and service context at runtime, and data activity within application and storage services. Together, these records provide a structured basis for operational investigation and control assurance.

BENEFITS FOR A BANK

Reduced blast radius.

Separating model, application and data contexts limits the reach and consequences of unexpected workload behaviour or configuration drift.

Stronger separation of use cases.

Models used for customer operations, legal, markets, risk or internal research can maintain distinct data paths, credentials and network authority.

Less configuration variance.

Platform-defined profiles give application teams a consistent foundation for authentication, routing, egress and runtime hardening.

More governable AI adoption.

Models, agents and generated applications operate inside a defined contract with clear ownership and operating scope.

Clearer assurance.

Security and risk teams can assess what enters, what can move, which inputs are trusted, where data resides and how additional requirements are authorised.

Preserved capability.

A workload remains free to act within its approved environment, while its authority is defined by an infrastructure boundary independent of probabilistic model behaviour.

TECHNICAL DETAIL · COMPARTMENT DESIGN FOR A BANK

Enclaves can be aligned to the bank's existing control domains. The selected unit should reflect where identity, data ownership, connectivity, change authority and recovery responsibility need to remain distinct.

Business unit | Legal entity | Data classification | Model risk tier | Environment | Use case

§ 04 · RATIONALE

Why enclaves are the right primitive for in-house AI.

Containers package and run software consistently. Kubernetes schedules and operates those workloads at scale. ACRA builds on both by adding a joined authority boundary for dynamic AI execution.

CONTAINER	Packages and runs software consistently.
KUBERNETES	Schedules and operates workloads at scale.
ACRA ENCLAVE	**Defines what a running workload is permitted to reach and do.**

Models and agents extend the traditional application model in four ways:

01 **Their behaviour is partly determined at runtime.** Tool calls, data access and generated actions can adapt as the workload operates.

02 **Their value comes from connectivity.** Models become operationally useful when connected to files, databases, APIs, code, browsers and internal systems.

03 **Their context can cross conventional application boundaries.** Derived information may be as sensitive as the source records from which it was produced.

04 **Their population can grow quickly.** AI-assisted development enables applications and workflows to scale beyond a bespoke, one-at-a-time governance model.

An enclave gives these systems a stable control foundation outside the application. As behaviour, connectors or code evolve, the environment continues to define the maximum authority available to the workload.

TECHNICAL DETAIL · DETERMINISTIC CONTROLS AROUND PROBABILISTIC SYSTEMS

Model output is probabilistic; network admission, identity validation, image policy and storage credentials are deterministic infrastructure controls. The enclave does not need to predict the model's next action. It defines which actions can resolve into permitted infrastructure operations.

This makes the enclave a stronger primitive for operating models in house because it is:

PRE-EXECUTION
Boundaries exist before the workload acts.

INDEPENDENT OF THE MODEL
Enforcement operates outside prompts, model alignment and probabilistic behaviour.

REPEATABLE
The same operating contract applies to many workloads.

COMPOSABLE
Applications receive the specific routes, data and tools appropriate to their function.

INDEPENDENTLY CONTROLLABLE
Connectivity and workload permissions are governed at layers outside the application.

COMPATIBLE
Extends Kubernetes-based environments alongside the bank's cloud, identity, data and security platforms.

THE PRINCIPLE

The workload can choose an action. The environment defines its authority.

ACRA makes that principle operational by turning the runtime environment into an enforceable, reviewable boundary for sensitive execution.

TECHNICAL DETAIL · RELATIONSHIP TO THE EXISTING STACK

ACRA operates as a Kubernetes-native governance layer. It uses the bank's underlying compute, networking and storage, integrates with identity and credential systems, and complements application security and monitoring. The enclave supplies the joined execution boundary across those layers, so each tool contributes to one coherent workload contract.

ENCLAVE · CUSTOMER-OPS · AGENT SESSION ● ILLUSTRATIVE

09:14:02	PERMIT	Read · case records · declared storage binding	DATA
09:14:03	PERMIT	Call · screening API · declared FQDN	NETWORK
09:14:05	DENY	Egress · undeclared external host · recorded	NETWORK
09:14:05	DENY	Read · markets enclave database · cross-enclave	DATA
09:14:07	DENY	Exec · pod in sealed namespace	ADMISSION
09:14:09	PERMIT	Write · case note · enclave-scoped credential	IDENTITY

FIG. 3 — Model behaviour can remain dynamic while the environment keeps available paths explicit and bounded.

§ 05 · NEXT STEP

Deployment approach.

Each implementation begins with a joint control mapping covering workloads, data classifications, identity sources, ingress and egress, availability and recovery, audit obligations, deployment environment and assurance evidence. This allows the enclave design to align with the bank's architecture, control framework and operating model from the outset.

TECHNICAL DETAIL · TYPICAL DESIGN OUTPUTS

The mapping produces the technical basis for implementation and assurance:

Enclave topology	Availability profile
Workload inventory	Backup and recovery design
Identity and trust paths	Policy-exception process
Ingress and egress matrix	Logging responsibilities
Data and credential map	Validation cases