Supporting Document

Evaluation Activities for collaborative PP-Module for Database-as-a-Service (DBaaS)

Version 0.5, 2026-07-20

Table of Contents

Chapter 1. Revision History

Table 1. Revision history

Version	Date	Description
0.1	2026-01-25	Initial draft of Evaluation Activities for DBaaS deployments.
0.2	2026-05-14	Aligned the SD with the CCiTC SaaS evaluation model: added the managed-service evaluation constraint, trusted-platform evidence handling, tenant-interface testing strategy, and per-SFR Evaluation Activities.
0.3	2026-06-30	Aligned with the v0.3 PP-Module: renamed iterated Base PP components to `/DBaaS`, added Evaluation Activities for `FMT_SMR.1/DBaaS`, `FMT_SMF.1/DBaaS`, and `FDP_BKP_EXT.1`, added explicit TOE/Platform/Trusted-Platform boundary enumeration guidance, and aligned references with cPP_DBMS Version 2.0 and the DBMS Cryptographic Functions Module Version 0.3. This SD remains in draft and is expected to mature on a later schedule than the Cloud and Crypto Module SDs.
0.4	2026-06-30	Refocused the FDP_BKP_EXT.1 Evaluation Activities on provider-plaintext-restore prevention and restore tenant-binding, with data-at-rest confidentiality of backups evaluated under the DBMS Cryptographic Functions Module SD. Version aligned with the v0.4 module and SD set.
0.4	2026-07-07	Managed-Service Evaluation Constraint contrast sentence updated: tenant-operated deployments cover both cloud-native and lift-and-shift models.
0.4	2026-07-07	Aligned SD sections with the module's component changes: resource governance activities moved to an FRU class section covering FRU_RSA.1/DBaaS and FRU_RSG_EXT.1; FIA_USB.1/DBaaS; FMT_MOF.1/DBaaS-Disable and -Modify; FAU_STG.2/DBaaS; FAU_SEL.1/DBaaS audit-query scoping noted as evaluated under FDP_ACF.1/DBaaS.
0.4	2026-07-08	Renamed the FPT_TUD_EXT.2 section to FPT_TUD_DBAAS_EXT.1 following the module's family rename.

Version	Date	Description
0.4	2026-07-14	Aligned with the module's maturity pass. FPT_TUD_DBAAS_EXT.1 activities rewritten for the Trusted Service Update design: TSS coverage of the declared deployment model, signed deployment artifacts, verification point, TSF/environment split of the update pipeline, and tenant version visibility; new signature-verification and tenant-version-visibility tests. Evaluator Access Strategy (provider-supported demonstration, recorded in the ETR) now invoked explicitly in the tests that require provider access: FDP_BKP_EXT.1 Test 1 (with an explicit ciphertext pass criterion), FMT_SMR.1/DBaaS Test 1, FMT_SMF.1/DBaaS Test 1, and FAU_STG.2/DBaaS Test 2. Added the OE.PROVIDER_ROLE_SEPARATION verification to Trusted Platform verification (distinct service and platform administrative populations, least privilege, corporate vertical integration permitted). Added Guidance Activities for FDP_RIP.1/DBaaS, FIA_USB.1/DBaaS, and FAU_SEL.1/DBaaS. Added the X.509 Functional Package reference.
0.4	2026-07-14	Structural completion to the Cloud/Crypto SD baseline. Added the Assurance Activities Overview: inherited SAR family mapping and DBaaS-specific considerations for ASE (managed-service ST verification, including the A.TRUSTED_PLATFORM_ADMIN population check), ADV (tenant-facing and provider service interface scope), ALC (continuous-deployment CM identification via the FPT_TUD_DBAAS_EXT.1.3 revision identifier, delivery reinterpreted as the authorized deployment path, ALC_FLR.3 coupled to trusted service update and bounded deferral), ATE (tenant-vantage testing, provider test evidence, revision-parity rule for non-production instances), and AVA (tenant-vantage vulnerability analysis and penetration testing with production-safety constraints); plus AGD_OPE and AGD_PRE documentation requirements for the service model. Added Introduction Structure and Terminology sections, a Handling Selections and Assignments section enumerating every open operation of the module, a complete Cross-Reference to CEM Work Units covering all SFR sections and the SAR considerations, and Test Environment Guidance for DBaaS Evaluators (two-tenant minimum, revocable external key, revision-verified demonstration instance, production-safety constraints).

Version	Date	Description
0.4	2026-07-14	Aligned with the module's two design corrections. FCS_CKM_DBAAS_EXT.1 activities now verify the tenant-authorization constraint on any claimed recovery mechanism (TSS and Guidance coverage; new Test 2 step exercising the mechanism and its audit record). FDP_ACC.2/DBaaS section renamed to FDP_ACC.1/DBaaS with a TSS check that the Tenant Isolation SFP is scoped as subset access control.
0.4	2026-07-18	Follow-on fixes from the audit-channel and vantage analysis. FIA_USB.1/DBaaS TSS activities verify tenant binding of subjects acting for automated autonomous management logic; FMT_SMF.1/DBaaS Test 1 gains a conditional step verifying an autonomous-action record identifies the autonomous subject with tenant attribution and tenant visibility. Audit Channel Identification verifies the per-channel retention and export treatment under P.AUDIT_RETENTION. FCS_CKM_DBAAS_EXT.1 recovery step may be performed jointly with FDP_BKP_EXT.1 Test 2, verifying restore tenant binding in the same run. Relationship to Other Documents names the two Base PP areas whose realization splits across planes: FIA (database authentication is TSF at V1; control-plane authentication via a cloud identity provider is verified under OE.EXTERNAL_SERVICES) and time (Base PP time activities do not apply to service event stream timestamps).
0.4	2026-07-18	Aligned with the module's service event stream audit channel. New Audit Channel Identification section in General Guidance: the evaluator verifies the ST's per-event channel assignment, the stream's implementing mechanism and boundary allocation, its tenant retrieval interface, and tenant visibility of provider-initiated records, before performing any audit-record check; channel substitution in either direction is not permitted. FCS_CKM_DBAAS_EXT.1 Test 2/2a and the recovery step now verify the enforcement-transition and recovery records in the ST-identified service event stream, retrievable while the database is not serving connections. FAU_STG.2/DBaaS scoped to the DBMS audit trail. ASE_TSS checks the channel identification is provided.

Version	Date	Description
0.4	2026-07-18	Added the Assurance Approach Rationale to the Assurance Activities Overview: the package substitutes requirement-bound adversarial testing for design-artifact depth (the SD's test legs are predominantly negative); at tenant vantage the reachable attack surface substantially coincides with the tested interface set; the certificate window matches the architectural cadence of the certified properties, with in-window operational change covered by continuously monitored Trusted Platform evidence and TSF change by ALC_FLR.3 plus the revision identifier; and the three assurance layers (algorithm validation, this PP-Configuration, cloud authorization) compose without duplication.
0.4	2026-07-18	Defined the four evidence vantages for managed-service evaluation (V1 tenant data plane, V2 tenant service control plane, V3 evaluator-controlled external services, V4 provider-internal operations) and added the informative Evidence Vantage Mapping appendix assigning vantages to every test activity, with scoping notes on where provider participation is required. The Evaluator Access Strategy now starts from this mapping and records deviations in the ETR. Relationship to Other Documents states the composition rule for the DBaaS PP-Configuration: Base PP and Crypto Module SD activities are executed from these vantages, with the Evaluator Access Strategy applied where they assume access the managed-service model does not provide. Provider-supported demonstration added to the managed-service evidence methods list.
0.4	2026-07-18	Aligned with the module's role-label clarification. Terminology states the global interpretation rule: role names in Evaluation Activities are the module's role labels, and the evaluator uses the TOE roles the ST maps to them. FMT_SMR.1/DBaaS TSS activities verify the mapping is present, unambiguous, and preserves the provider/tenant authority boundaries; Test 1 verifies the mapped TOE roles under their TOE names; ASE_TSS checks the mapping is provided.

Version	Date	Description
0.4	2026-07-18	Aligned with the module's key-state enforcement interval assignment in FCS_CKM_DBAAS_EXT.1.3. Handling Selections and Assignments now lists the interval with its permitted-completion constraints. TSS activities require the key-state detection mechanism, any unreachability grace period, and the derivation of the completed interval from detection latency, grace period, and cached-key lifetime; Guidance activities require the interval and post-expiration service state to be documented. Test 2 rewritten to verify denial no later than interval expiration with the enforcement-state transition audit record; new Test 2a exercises key-provider unreachability against the documented grace behavior, with a provider-evidence fallback where connectivity cannot be interrupted.
0.4	2026-07-20	Aligned customer-controlled key evaluation with the Crypto Module's role-specific ML-KEM design. Added TSS and guidance checks for the "Established through Key Encapsulation" origin, TOE-versus-external operation allocation, KEM-to-Master-Key derivation, and ongoing customer authorization. Added Test 4 and evidence-vantage mapping to verify that a KEM-established Master Key remains subject to the bounded customer revocation requirement; algorithm and integration testing remains in the Crypto Module SD.
0.5	2026-07-20	Rolled the completed managed-service Evaluation Activities, evidence-vantage model, provider-supported demonstration paths, and Crypto Module coordination into the Version 0.5 review set.

Chapter 2. Introduction

2.1. Technology Area and Scope of Supporting Document

This Supporting Document (SD) defines Evaluation Activities (EAs) for the **collaborative PP-Module for Database-as-a-Service** (DBMS_MOD_DBAAS), version 0.5. The technology area addressed is a Database Management System (DBMS) offered as a managed Database-as-a-Service, in which the cloud service provider operates and manages the DBMS on behalf of tenants.

This SD provides Evaluation Activities that evaluators shall perform to determine whether a TOE satisfies the Security Functional Requirements (SFRs) specified in the DBaaS PP-Module. These activities complement the Evaluation Activities defined in the Supporting Document for the collaborative Protection Profile for Database Management Systems (cPP_DBMS SD), which remains applicable for all SFRs inherited from the Base PP, and the Supporting Document for the DBMS Cryptographic Functions Module, which provides cryptographic, data-at-rest, data-in-transit, and key-management Evaluation Activities.

The Evaluation Activities in this SD are derived from the Common Evaluation Methodology (CEM:2022) work units and are aligned with the CCiTC Guidance for Cloud Evaluations Software-as-a-Service (SaaS) model.

2.2. Structure of the Document

This SD is organized as follows:

- **Introduction**: scope, the managed-service evaluation constraint, the evidence vantages (V1–V4) from which test evidence is collected, terminology, and document relationships;
- **General Guidance for Evaluators**: Trusted Platform verification (including provider role separation), boundary enumeration, the Evaluator Access Strategy, audit channel identification, tenant isolation testing strategy, external service dependencies, and the handling of selections and assignments;
- **Assurance Activities Overview**: the inherited SAR families and the DBaaS-specific considerations for ASE, ADV, ALC, ATE, and AVA, together with the DBaaS documentation requirements for AGD_OPE and AGD_PRE;
- **Per-SFR Evaluation Activities** (FCS, FDP, FRU, FIA, FMT, FAU, FPT classes): TSS, Guidance, and Test activities for each SFR of the PP-Module;
- **Appendices**: cross-reference to CEM work units, the evidence vantage mapping for every test activity, test environment guidance, and document references.

2.3. Terminology

Terms defined in the Glossary of the DBaaS PP-Module [[DBMS_MOD_DBAAS]] (Customer-Controlled Key, Service Provider Administrator, Tenant, Tenant Administrator, Tenant User, Trusted

Platform, TOE Platform) are used in this SD with the same meanings. The following evaluator-facing terms are additionally used:

Tenant context

An evaluator-provisioned tenant of the service, together with its users, roles, data, and tenant-scoped resources, used as a test subject. The Tenant Isolation Testing Strategy requires at least two distinct tenant contexts.

Provider-supported demonstration

An evaluation activity performed by or with the service provider under the Evaluator Access Strategy, in which the evaluator directs and observes the exercise of functionality that requires provider access, and records the method and results in the ETR.

Service software revision

The state of the deployed DBMS service software identified by the version or revision identifier of FPT_TUD_DBAAS_EXT.1.3. For continuously deployed services this identifier takes the place of a traditional version number for CM identification, testing parity, and flaw-remediation verification.

Role names in the Evaluation Activities of this SD — Tenant User, Tenant Administrator, Service Provider Administrator — are the module-level role labels of `FMT_SMR.1/DBaaS`. Per that component's application note, the TOE is not required to name its roles verbatim; the ST maps each label to the TOE's own role terminology. Wherever an activity directs the evaluator to act as, provision, or observe one of these roles, the evaluator shall use the TOE role or roles the ST maps to that label.

2.4. Managed-Service Evaluation Constraint

Unlike a tenant-operated cloud deployment (cloud-native or lift-and-shift), the DBaaS TOE is a provider-operated managed service. The evaluator typically does not have access to:

- the underlying operating system;
- the hypervisor or physical hardware;
- raw process memory;
- provider operational tooling; or
- Service Provider administrative accounts.

Evaluation Activities in this SD are therefore designed to be performed through:

1. **Tenant interfaces**: standard SQL clients, tenant-facing web consoles, APIs, audit views, and management interfaces accessible to the customer.
2. **Documentation review**: analysis of the ST, TSS, operational guidance, and provider architecture descriptions.
3. **Scheme-accepted evidence**: cloud authorization, certification, scheme policy, or other evidence accepted by the applicable evaluation scheme for Trusted Platform assumptions and operational environment objectives.
4. **Provider-supported demonstration**: exercise of provider-restricted functionality directed and

observed by the evaluator under the Evaluator Access Strategy, recorded in the ETR.

2.5. Evidence Vantages

Test evidence for this SD — and, when the DBaaS PP-Configuration is evaluated, for Base PP and Crypto Module activities executed against the managed service (see Section 2.6, "Relationship to Other Documents") — is collected from four vantages. The vantage identifiers V1 through V4 are used throughout this SD, and the Evidence Vantage Mapping appendix assigns them to every test activity.

V1 — Tenant data plane

The database interfaces of an evaluator-provisioned tenant context — SQL clients, drivers, and database-level APIs — exercised as the TOE roles mapped to the Tenant User and Tenant Administrator labels. This is the vantage of the Base PP's ordinary user and administrator activities.

V2 — Tenant service control plane

The tenant-facing management surface of the cloud service — service consoles, management APIs, and command-line tools available to the evaluation tenancy — used to provision tenant contexts, configure customer-controlled keys, initiate backup and restore, configure resource quotas and audit export where tenant-visible, and query the service software revision. This vantage is evaluator-driven and requires no provider involvement beyond the evaluation tenancy itself. Which functions appear at V2 rather than V4 varies by TOE and is settled by the ST's interface identification and role-label mapping.

V3 — Evaluator-controlled external services

Services outside the TOE but under evaluator control, on which tests depend as instruments: the external key management service holding the revocable test key (through which key revocation and key-provider unreachability are induced) and the external audit sink (at which audit export is observed).

V4 — Provider-internal operations

Functions exercisable only by the service provider: provider administrative roles, the update deployment pipeline, restore paths not exposed at V2, and fleet operations. This vantage is never directly accessible to the evaluator; its evidence is obtained through provider-supported demonstration, provider test evidence, or provider documentation under the Evaluator Access Strategy, with every such reliance recorded in the ETR.

A test whose primary vantage is V1–V3 is executed by the evaluator without provider involvement; a test with a V4 leg requires provider participation that the laboratory and provider should schedule when scoping the evaluation.

2.6. Relationship to Other Documents

This SD is used in conjunction with:

- **cPP_DBMS SD**, Version 2.0: Evaluation Activities for SFRs inherited from the Base PP.

- **DBMS Cryptographic Functions Module SD**, Version 0.5: cryptographic, data-at-rest, data-in-transit, key-management, and algorithm-validation Evaluation Activities.
- **DBaaS PP-Module**, version 0.5: requirements definition.

Where SFRs are iterated by the PP-Module as `/DBaaS`, the Evaluation Activities in this SD supplement those in the Base PP SD for the tenant-scoped refinements.

When the DBaaS PP-Configuration is evaluated, the Base PP SD and Crypto Module SD activities are executed against the same provider-operated managed service as the activities of this SD. Those documents were written for TOEs where the evaluator holds full administrative access; under this PP-Configuration their activities are executed from the vantages defined in Section 2.5, "Evidence Vantages". A Base PP or Crypto Module activity exercisable through the tenant data plane (V1) or tenant service control plane (V2) is performed there unchanged. Where such an activity assumes host-level, operating-system, or full-DBA access that the managed-service model does not provide to tenants, the evaluator shall apply the Evaluator Access Strategy (Section 3.3, "Evaluator Access Strategy") — determining whether the SD in question permits reliance on TSS review, operational guidance, scheme-accepted evidence, or a provider-supported demonstration (V4) — and shall record each such reliance in the ETR.

Two Base PP areas warrant naming, because their realization splits across planes in a managed service. Identification and authentication: the Base PP's FIA activities apply to the database authentication interfaces of the TOE (V1); authentication to the tenant service control plane (V2) is typically performed by a cloud identity provider, which is an external service under OE.EXTERNAL_SERVICES and is verified as an external service dependency rather than under the Base PP FIA activities — the ST's boundary enumeration identifies which authentication surfaces are TSF. Time: audit timestamps for DBMS audit trail records derive from the time source the Base PP activities verify, while service event stream records derive from the Trusted Platform's time service under OE.EXTERNAL_SERVICES; the evaluator relies on the ST's identification of each and does not apply the Base PP time activities to the stream.

Chapter 3. General Guidance for Evaluators

3.1. Verifying the Trusted Platform

The evaluator shall review the ST and operational guidance to identify the Trusted Platform relied upon by the TOE. The evaluator shall verify that the ST identifies the cloud authorization, certification, scheme policy, or other evidence submitted to support the Trusted Platform assumptions and Operational Environment objectives. The evaluator shall confirm that the stated scope of the evidence covers the identified Cloud Service Offering, region, services, and evaluated configuration, and shall document this reliance in the ETR.

The evaluation authority determines whether the submitted evidence is acceptable under the applicable scheme policy, including recognized authorization programs, assurance levels, currency, and exceptions. The evaluator does not independently establish Trusted Platform acceptance criteria or treat cloud authorization evidence as satisfying a TOE SFR unless the applicable requirement expressly permits use of a platform-provided mechanism.

Provider role separation (OE.PROVIDER_ROLE_SEPARATION). The evaluator shall verify that the provider's role and account model documentation demonstrates that administrative authority over the DBaaS service and administrative authority over the Trusted Platform are exercised through logically distinct roles, accounts, and control planes, assigned according to least privilege, such that no single administrative identity holds both. Vertical integration — the same provider operating both the service and the platform — is permitted; the verification addresses the administrative populations and their control planes, not corporate structure. Where the documentation alone is not sufficient, the evaluator shall use a provider-supported demonstration of the role and account model (for example, enumeration of the distinct administrative directories, IAM domains, or control planes) under the Evaluator Access Strategy, and shall record the evidence relied upon in the ETR.

3.2. TOE, TOE Platform, and Trusted Platform Boundary Enumeration

Because a DBaaS architecture may allocate components and supporting services differently, the evaluator shall confirm that the boundary between TOE-enforced behavior and platform-provided properties is explicitly enumerated and is consistent with all claimed SFRs, assumptions, and Operational Environment objectives. Provider ownership or operation of a component does not by itself determine its allocation.

The evaluator shall:

1. Verify that the TSS enumerates, for each claimed SFR, the security behavior performed by the TOE and any supporting property relied upon from the TOE Platform, Trusted Platform, or another operational environment component.
2. Verify that a claimed SFR is not allocated wholly to the Trusted Platform or operational environment unless the applicable requirement expressly permits use of a platform-provided mechanism. Where such use is permitted, verify that the TSS distinguishes the evaluated TSF

behavior from the external property on which it relies.

3. For each external reliance, verify that the allocation is supported by operational guidance and is linked to an assumption or Operational Environment objective. Where the TOE behavior at the boundary is testable through tenant-accessible interfaces, apply the relevant test activities in this SD.
4. Record in the ETR each external reliance, the applicable assumption or Operational Environment objective, the identified evidence, and the applicable scheme policy or acceptance decision.

This enumeration is the primary mechanism by which the TOE/Trusted-Platform boundary is made explicit and evaluable; the per-SFR TSS activities below compel the enumeration, and the test activities verify it wherever the evaluated configuration permits.

3.3. Evaluator Access Strategy

The evaluator shall identify the interfaces available in the evaluated configuration and map each test to one or more tenant-accessible or provider-supported evaluation interfaces. The Evidence Vantage Mapping appendix (Appendix B, *Evidence Vantage Mapping*) provides the expected vantage assignment for every test activity of this SD; the evaluator shall use it as the starting point for this mapping and shall record in the ETR any deviation driven by the evaluated configuration — in particular, functions the mapping places at the tenant service control plane (V2) that the evaluated TOE exposes only to provider roles (V4), or the reverse. When a direct test is not feasible because the function is outside the TOE boundary or requires provider-only access, the evaluator shall determine whether the SD permits reliance on TSS review, operational guidance, scheme-accepted evidence, or a provider-supported demonstration.

3.4. Audit Channel Identification

The module's auditable-events refinement assigns each auditable event to the DBMS audit trail or to the service event stream (or, for events whose plane varies by TOE, to either as identified in the ST). Before performing any test activity that verifies an audit record, the evaluator shall verify that the ST identifies:

- for each module auditable event, the channel that carries it, consistent with the Audit Channel column of the module's auditable-events table;
- the mechanism that implements the service event stream, its allocation to the TOE, TOE Platform, or Trusted Platform, and the tenant interface through which its records are retrieved (typically at the tenant service control plane, V2);
- how service event stream records concerning a tenant's service instance, customer-controlled keys, or backups are made visible to that tenant regardless of the acting subject, including provider-initiated and service-initiated actions;
- the retention and export treatment of each channel under P.AUDIT_RETENTION: which DBMS audit trail records and which service event stream records are exportable to tenants, and whether any claimed audit export function covers service event stream records; and
- where the service event stream is delivered by the Trusted Platform, the reliance on

OE.TRUSTED_PLATFORM and OE.EXTERNAL_SERVICES for its integrity and availability, recorded in the ETR with the Trusted Platform verification.

Each per-SFR test activity of this SD that checks for an audit record is performed against the channel the ST identifies for that event. An audit record required in the service event stream is not satisfied by a record in the DBMS audit trail that becomes unreadable when the database stops serving connections, and the reverse substitution is likewise not permitted.

3.5. Tenant Isolation Testing Strategy

Testing for logical isolation requires the evaluator to provision at least two distinct tenant contexts, such as separate tenants, databases, pluggable databases, schemas, namespaces, or accounts, depending on the ST claim. The evaluator shall attempt cross-tenant access through the same interfaces available to the tenant in the evaluated configuration.

The evaluator shall avoid relying on undocumented provider privileges as primary test mechanisms unless those privileges are part of the evaluated configuration or are used only to set up or observe the test under controlled conditions.

3.6. External Service Dependencies

When the TOE relies on external key management, identity provider, logging, object storage, backup, management, or trusted time services, the evaluator shall verify that the ST identifies:

- whether the service is part of the TOE, TOE Platform, Trusted Platform, or operational environment;
- which SFRs, assumptions, and operational environment objectives depend on the service;
- what evidence supports the service's claimed security properties; and
- which operational guidance configures the service for the evaluated configuration.

3.7. Handling Selections and Assignments

The PP-Module leaves the following operations open for completion by the ST author. For each, the evaluator shall verify that the ST completes the operation with permitted content, that the completion is consistent with the TSS, and that the corresponding test activities of this SD exercise the completed value rather than the generic placeholder:

- `FCS_CKM_DBAAS_EXT.1.1` — the customer-controlled key mechanism selection, which shall couple to the Crypto Module's Key Origin selection as required by the module's application note;
- `FCS_CKM_DBAAS_EXT.1.3` — the assigned key-state enforcement interval, which shall be a fixed documented bound covering worst-case key-state detection latency, any unreachability grace period, and the lifetime of any cached key material, as required by the module's application note (an indefinite interval, or one extensible by service provider action, is not a permitted completion);
- `FDP_RIP.1.1/DBaaS` — the shared-resource selection and any assigned additional resources;

- FDP_ACF.1.3/DBaaS and FDP_ACF.1.4/DBaaS — the assigned explicit-authorization and explicit-denial rules (an assignment completed as "none" is acceptable);
- FRU_RSA.1.1/DBaaS — the controlled-resource selection, any assigned additional resources, and the simultaneity selection;
- FRU_RSG_EXT.1.1 — the enforcement-action selection or assigned action;
- FMT_SMR.1.1/DBaaS — the selection of automated autonomous management logic and any assigned additional roles;
- FMT_SMF.1.1/DBaaS — any assigned additional management functions;
- FMT_MOF.1.1/DBaaS-Modify — the assigned additional functions and the authorized-role selection;
- FPT_TUD_DBAAS_EXT.1.1 — the deployment-model selection and, where tenant-configurable deferral is selected, the assigned maximum deferral period.

Where a completed selection makes an activity of this SD inapplicable (for example, deferral tests when uniform rolling deployment is selected), the evaluator shall record the inapplicability and its basis in the ETR rather than omitting the activity silently.

Chapter 4. Assurance Activities Overview

4.1. Assurance Approach Rationale

The assurance package is EAL2 augmented by ALC_FLR.3, and the number understates where the assurance comes from. EAL2 bounds the design evidence — interface specification and basic subsystem design, without implementation representation — but confidence in the TOE's security behavior is produced by this SD's Evaluation Activities, and those are predominantly negative: cross-tenant access is attempted through every documented interface and denied; metadata, audit, and residual-data isolation are probed from a second tenant context; plaintext key export is attempted through every tenant-facing interface and denied; key revocation is enforced within the completed key-state enforcement interval; plaintext restore is attempted against an explicit ciphertext pass criterion; unauthorized quota modification, tenant-binding manipulation, audit tampering, and the disabling of audit or encryption are each attempted and denied; an invalid update signature is presented and rejected. The package substitutes requirement-bound adversarial testing for design-artifact depth.

That substitution is unusually effective for this TOE type. Against locally installed software an attacker holds the binary, the file system, and process memory, so interface-level testing exercises a modest fraction of the reachable attack surface — closing the remainder is what implementation-level design assurance exists for. A tenant of a provider-operated DBaaS reaches the TOE only through the documented tenant interfaces: SQL, programmatic APIs, and the tenant service control plane. The tenant-reachable attack surface therefore substantially coincides with the interface set these Evaluation Activities exercise, and AVA_VAN.2 penetration testing probes the same vantage for paths the documentation does not admit.

The evaluation is point-in-time, and the certificate window is matched to what is being certified. The properties this PP-Module evaluates — the tenant-isolation architecture, the customer-controlled key hierarchy, the restore path, the update mechanism — are architectural, and change on a cadence no faster than scheme certificate-validity windows. Change that does occur inside a validity window divides into two streams, each with its own assurance: operational change (configuration, platform, personnel) belongs to the Trusted Platform, whose accepted evidence — cloud authorization under frameworks such as FedRAMP, IRAP, or C5 — carries the continuous operational monitoring of the production environment that CC assurance continuity, being change-driven and evaluator-mediated, does not; TSF change is governed by ALC_FLR.3 and the trusted service update requirement, whose tenant-queryable revision identifier identifies every deployed change and supplies the CM identification on which assurance continuity depends. Nor is the certificate unattended between evaluations: schemes engage listed products on major vulnerabilities, require impact analysis or a remediation plan, and reserve delisting for non-response. The residual difference from authorization-framework continuous monitoring is cadence and object — a standing, scheduled evidence flow about the operating environment versus event-driven oversight of the product — and the standing flow is precisely what the consumed Trusted Platform evidence supplies. This division follows the object of each regime, not a limit of the CC language: the ALC class carries development-environment analogues of much of the operational control surface (ALC_DVS for physical and personnel security, ALC_CMC and ALC_CMS for change control, ALC_DEL for delivery, ALC_FLR for flaw handling), and site-certification practice shows those families can be audited in depth. This PP-Configuration claims the EAL2 subset of that class

and consumes the cloud authorization for the operational surface instead — a non-duplication choice, not a capability gap.

The composition is deliberate and non-duplicative. Cloud authorization frameworks assess whether the provider organization operates the service responsibly; they do not functionally test whether a security mechanism is correct — no assessor under those frameworks provisions two tenants and attempts cross-tenant access, revokes a key and times enforcement, or attempts a plaintext restore. Algorithm validation (CAVP, CMVP, and the CCDB-018 catalogue, consumed through the mandatory DBMS Cryptographic Functions Module) assesses cryptographic correctness but not service behavior. This PP-Configuration supplies the layer between them: independent, specification-bound demonstration that the service's tenant-facing security mechanisms behave as claimed. Each layer is consumed, not duplicated — algorithm validation enters through the Crypto Module, and cloud authorization enters as scheme-accepted Trusted Platform evidence.

4.2. Mapping to Base PP SARs

The DBaaS PP-Module inherits all Security Assurance Requirements (SARs) from the collaborative Protection Profile for Database Management Systems (cPP_DBMS): EAL2 augmented by ALC_FLR.3. No additional or modified SARs are introduced by this PP-Module. The provider-operated managed-service model does, however, change how several inherited SAR families are exercised; those considerations are defined in this section.

4.2.1. Inherited SAR Families

SAR Family	Components	Primary SD Coverage
ADV (Development)	ADV_ARC.1, ADV_FSP.2, ADV_TDS.1	cPP_DBMS SD + managed-service interface considerations in this SD
AGD (Guidance Documents)	AGD_OPE.1, AGD_PRE.1	cPP_DBMS SD + DBaaS-specific documentation requirements in this SD
ALC (Life-cycle Support)	ALC_CMC.2, ALC_CMS.2, ALC_DEL.1, ALC_FLR.3	cPP_DBMS SD + continuous-deployment and flaw-remediation considerations in this SD
ASE (Security Target Evaluation)	ASE_CCL.1, ASE_ECD.1, ASE_INT.1, ASE_OBJ.2, ASE_REQ.2, ASE_SPD.1, ASE_TSS.1	cPP_DBMS SD + PP-Module conformance and allocation verification in this SD
ATE (Tests)	ATE_COV.1, ATE_FUN.1, ATE_IND.2	cPP_DBMS SD + managed-service testing strategy in this SD
AVA (Vulnerability Assessment)	AVA_VAN.2	cPP_DBMS SD + tenant-vantage vulnerability analysis in this SD

4.2.2. Applying Base PP SAR Activities

For each SAR inherited from the cPP_DBMS, the evaluator shall:

1. Perform all Evaluation Activities specified in the cPP_DBMS SD;
2. Apply the managed-service considerations from this SD where applicable, using the Evaluator Access Strategy (Section 3.3, “Evaluator Access Strategy”) where an activity requires provider access; and
3. Document findings, including every reliance on provider-supported demonstration or provider documentation, in the Evaluation Technical Report (ETR).

4.3. DBaaS-Specific Assurance Activities

4.3.1. ASE: Security Target Evaluation

In addition to the Base PP ST evaluation activities, the evaluator shall verify:

ASE_INT (ST Introduction):

- The TOE description accurately describes the provider-operated managed-service architecture and identifies the tenant-facing database, administrative, and programmatic interfaces;
- The TOE boundary distinguishes the TOE from the TOE Platform and Trusted Platform consistently with the module’s Security Function Allocation requirements;
- The TOE reference identifies the evaluated service software by the same version or revision identifier that tenants can query under FPT_TUD_DBAAS_EXT.1.3.

ASE_CCL (Conformance Claims):

- The PP-Configuration is exactly `cPP_DBMS + DBMS DBaaS Module + DBMS Cryptographic Functions Module`, and conformance is claimed as Exact Conformance;
- The DBMS in the Cloud Module is not claimed (the modules are mutually exclusive);
- The FCS_CKM_DBAAS_EXT.1.1 mechanism selections couple correctly to the Crypto Module’s Key Origin selections as required by the module’s application note.

ASE_SPD (Security Problem Definition):

- All threats, assumptions, and organizational security policies of the PP-Module are included;
- The ST does not extend A.TRUSTED_PLATFORM_ADMIN to the Service Provider Administrator population (this would contradict OE.PROVIDER_ROLE_SEPARATION and vacate T.KEY_DISCLOSURE_TO_PROVIDER_ADMIN).

ASE_OBJ (Security Objectives):

- TOE and OE objectives trace to the security problem definition as defined in the module, including OE.PROVIDER_ROLE_SEPARATION;
- The ST identifies how each OE objective is upheld in the evaluated configuration, including the evidence intended to demonstrate provider role separation.

ASE_TSS (TOE Summary Specification):

- The TSS provides the per-SFR TOE/environment enumeration required by Section 3.2, “TOE, TOE

Platform, and Trusted Platform Boundary Enumeration";

- The TSS provides the role-label mapping required by the module's FMT_SMR.1/DBaaS application note, on which the role references throughout this SD's activities depend;
- The ST provides the audit channel identification required by Section 3.4, "Audit Channel Identification", including the service event stream mechanism, its boundary allocation, and its tenant retrieval interface;
- The TSS describes the end-to-end update process and the customer-controlled key architecture at the level of detail required by the per-SFR activities of this SD.

4.3.2. ADV: Development

The Base PP's ADV_ARC.1, ADV_FSP.2, and ADV_TDS.1 activities apply to the DBMS service. The evaluator shall additionally verify:

- The functional specification covers the tenant-facing database, administrative, and programmatic interfaces and the provider service administration interfaces that terminate at the TOE; interfaces internal to the Trusted Platform are outside the functional specification's scope and shall not be used to satisfy it;
- The security architecture description explains how the TSF preserves tenant separation and TSF self-protection in a shared, provider-operated service, consistent with the Tenant Isolation SFP;
- Where design evidence describes provider-internal mechanisms not observable from tenant interfaces, the evaluator relies on the developer's design documentation and provider-supported demonstration under the Evaluator Access Strategy, recording the reliance in the ETR.

4.3.3. ALC: Life-cycle Support

The managed-service model changes how the ALC families are exercised. The evaluator shall apply the Base PP ALC activities with the following interpretations:

ALC_CMC.2 / ALC_CMS.2 (Configuration Management):

- CM evidence shall uniquely identify the deployed service software. For continuously deployed services, the CM reference is the version or revision identifier of FPT_TUD_DBAAS_EXT.1.3: the evaluator shall verify that the identifier queryable by tenants corresponds to a CM-tracked configuration item, so that the evaluated TOE is identifiable in production;
- The CM scope includes the deployment artifacts to which signature verification applies under FPT_TUD_DBAAS_EXT.1.2.

ALC_DEL.1 (Delivery):

- In a provider-operated service there is no delivery of the TOE to the consumer; the delivery procedure is the provider's deployment path from CM-controlled artifacts to the running service. The evaluator shall verify that the documented delivery procedure is the authorized update process evaluated under FPT_TUD_DBAAS_EXT.1, including its signature verification, and shall not require consumer-side delivery evidence that the service model cannot produce.

ALC_FLR.3 (Systematic Flaw Remediation):

- The flaw remediation procedures shall connect to the trusted service update mechanism: remediation of a security flaw reaches tenants through the update path of FPT_TUD_DBAAS_EXT.1, and the evaluator shall verify the procedures identify remediation timelines, the tenant notification or advisory mechanism, and how a tenant determines — via the version or revision identifier — that a remediated service software revision is serving them;
- Where the deployment model includes tenant-configurable deferral, the evaluator shall verify the flaw remediation procedures address security-critical flaws in relation to the maximum deferral period.

4.3.4. ATE: Tests

ATE_COV.1 / ATE_FUN.1 (Developer Testing):

- Developer test evidence may cover provider-restricted functions that the evaluator cannot exercise directly; the evaluator shall verify that the developer's test coverage includes those functions and shall use that evidence, together with provider-supported demonstration, when applying the corresponding test activities of this SD.

ATE_IND.2 (Independent Testing):

- Independent testing is conducted from the tenant vantage through the interfaces of the evaluated configuration, per the Managed-Service Evaluation Constraint (Section 2.4, "Managed-Service Evaluation Constraint") and the Tenant Isolation Testing Strategy (Section 3.5, "Tenant Isolation Testing Strategy");
- Where a test requires provider access, the evaluator shall use provider-supported demonstration under the Evaluator Access Strategy and record the method and results in the ETR;
- Testing may be performed on a non-production instance of the service provided the evaluator verifies, using the FPT_TUD_DBAAS_EXT.1.3 identifier or CM evidence, that the tested instance runs the same service software revision as the evaluated production configuration;
- The evaluator shall confirm that the audit records relied upon by the test activities of this SD are generated as required by the module's auditable-events refinement of the Base PP's Table 4.

4.3.5. AVA: Vulnerability Assessment

AVA_VAN.2 (Vulnerability Analysis):

The evaluator's public-domain vulnerability search shall include:

- published vulnerabilities in the DBMS engine lineage underlying the service, including vulnerabilities disclosed against non-service distributions of the same engine;
- published vulnerabilities and advisories for the managed service itself; and
- vulnerability classes specific to multi-tenant managed services: cross-tenant isolation escape through SQL, metadata, or diagnostic channels; tenant-context confusion in session or connection handling; audit-scope leakage; resource-exhaustion attacks on shared components;

key-reference confusion or authorization bypass in customer-controlled key integration; and abuse of restore or export paths.

Penetration testing shall:

- be conducted from the tenant vantage, using both least-privileged tenant user and Tenant Administrator contexts, against evaluator-provisioned tenants only — the evaluator shall never direct isolation or exhaustion testing at tenants not provisioned for the evaluation;
- focus on the TOE's tenant-facing attack surface; the Trusted Platform itself is not the subject of penetration testing, but the evaluator shall attempt to reach Trusted Platform assumptions through TOE interfaces (for example, attempting to obtain platform credentials, metadata-service access, or provider-role capabilities from a tenant context) and verify that such attempts fail;
- respect the production-safety constraints of Appendix C, *Test Environment Guidance for DBaaS Evaluators*, coordinating resource-exhaustion and update-path testing with the provider and preferring a non-production instance at the same service software revision.

4.4. Documentation Requirements

4.4.1. AGD_OPE: Operational User Guidance — DBaaS-Specific Requirements

The operational user guidance shall include managed-service sections addressing:

Shared-Responsibility Description:

- Which security functions are operated by the provider and which are configured or exercised by the tenant, consistent with the ST's Security Function Allocation;
- The tenant-facing interfaces of the evaluated configuration.

Customer-Controlled Key Operations:

- Configuring customer-controlled keys (import, external KMS integration, key references, or customer-authorized ML-KEM establishment) per the coupled Crypto Module Key Origin;
- Key rotation, revocation, and disabling of the TOE's authorization to use a key;
- Expected TOE behavior on revocation and any documented recovery mechanism, including the tenant authorization its use requires.

Audit Operations:

- Tenant-scoped audit query, audit selection configuration, and audit export;
- The tenant-visible audit events for the module's SFRs.

Resource Governance:

- Configuring tenant resource quotas and enforcement behavior, and monitoring quota enforcement.

Service Updates:

- Querying the current service version or revision identifier;
- Update notifications, deferral options or maintenance windows where claimed, and the maximum deferral period.

4.4.2. AGD_PRE: Preparative Procedures — Service Provisioning

For a provider-operated service, preparative procedures address how a tenant provisions and configures the service into the evaluated configuration. The preparative procedures shall include:

Provisioning:

- How to subscribe to or provision the service such that the resulting instance corresponds to the evaluated configuration (service tier, region or deployment options, and security-relevant provisioning parameters);
- How to verify that the provisioned service is the evaluated TOE, using the version or revision identifier of FPT_TUD_DBAAS_EXT.1.3.

Initial Secure Configuration:

- Enabling and configuring customer-controlled keys before placing protected data in the service;
- Configuring tenant roles, audit export, and resource governance to the evaluated configuration.

Verification:

- Confirming, through tenant-accessible status and audit interfaces, that the initial configuration matches the evaluated configuration.

The evaluator shall verify these guidance elements are present and accurate by applying the documented procedures when provisioning the evaluation tenants.

Chapter 5. FCS Class: Cryptographic Support

5.1. FCS_CKM_DBAAS_EXT.1 Customer-Controlled Keys

5.1.1. Evaluation Activities

5.1.1.1. TSS Activities

The evaluator shall verify the TSS describes:

- the customer-controlled key management architecture;
- the supported customer-controlled key models, such as BYOK, HYOK, external KMS integration, key references, key handles, or customer-authorized ML-KEM establishment;
- the boundary between TOE key-management logic and external key management services;
- the protocols and authentication mechanisms used for external key management service access, including the trusted channel specified by the DBMS Cryptographic Functions Module and applicable Functional Packages;
- whether plaintext customer-controlled master key material is imported into, stored by, cached by, wrapped by, transformed by, or re-exported by the TOE;
- how the TOE prevents service provider administrators from viewing, exporting, recovering, or otherwise obtaining plaintext customer-controlled master key material through TOE interfaces;
- the applicable DBMS-specific and Catalogue-derived components included through the DBMS Cryptographic Functions Module for key import, key wrapping, key destruction, and data-at-rest encryption, together with Functional Package components for trusted channels;
- when ML-KEM establishment is selected, the mapping to the Crypto Module's "Established through Key Encapsulation" origin; the TOE role in `FCS_COP.1/KeyEncap`; the optional local-KeyGen claim in `FCS_CKM.1/KEM`; the `FCS_CKM.5` Master Key derivation and context; raw shared-secret destruction; and every operation allocated to an external KMS or customer-controlled service; and
- the TOE behavior when authorization to use the applicable customer-controlled master key is revoked, unavailable, or denied;
- the mechanism by which the TOE detects key-state changes at the external key provider (for example, a polling interval or an event notification channel), any grace period during which the TOE continues operating on cached key material while the external key provider is unreachable, and how the completed key-state enforcement interval of FCS_CKM_DBAAS_EXT.1.3 covers the sum of worst-case detection latency, any such grace period, and the lifetime of cached key material; and
- where a recovery mechanism is claimed under FCS_CKM_DBAAS_EXT.1.3, how its use requires authorization by the tenant or an agent under the tenant's control, and how that use is audited. A recovery mechanism exercisable by the service provider without tenant authorization does not satisfy the requirement.

If the TOE caches key material, key handles, or derived keys after successful KMS authorization, the

evaluator shall verify that the TSS identifies the cache duration, protection mechanism, revocation behavior, and conditions under which cached material is invalidated, and that the cache duration is consistent with the completed key-state enforcement interval.

When customer-authorized ML-KEM establishment is selected, the evaluator shall verify that the customer or its authorized external entity retains a control whose revocation, unavailability, or denial is enforced within the assigned key-state enforcement interval. A one-time KEM exchange after which the provider can use the derived Master Key indefinitely without further customer authorization does not satisfy the module's customer-controlled model. Algorithm correctness, KEM role testing, derivation-context testing, and invalid-ciphertext behavior are evaluated under the Crypto Module SD and are not duplicated here.

5.1.1.2. Guidance Activities

The evaluator shall verify the operational guidance describes:

- how a customer configures customer-controlled keys for the TOE;
- how a customer revokes or disables the TOE's authorization to use an external key;
- expected TOE behavior after key authorization is revoked, unavailable, or denied, including the documented key-state enforcement interval, any unreachability grace period within it, and the service state the TOE enters when the interval expires;
- any documented recovery mechanism that permits protected data access after revocation, including the tenant authorization its use requires; and
- any customer-visible audit or status information related to key use, key failures, or key authorization.

For customer-authorized ML-KEM establishment, the evaluator shall additionally verify that the guidance describes the customer-controlled peer or key configuration, how the customer authorizes establishment, how the customer revokes or denies continued use, the bounded enforcement behavior, and which ML-KEM operations are performed by the TOE versus an external service.

5.1.1.3. Test Activities

Test 1: Provider Inaccessibility Verification

1. Configure the TOE to use a customer-controlled master key or external key reference as described in the operational guidance.
2. Access protected database data as an authorized tenant user or tenant administrator.
3. Verify that access succeeds.
4. Attempt to export, view, recover, or display the plaintext master key material through each tenant-facing DBMS interface and management interface available in the evaluated configuration.
5. Verify that each attempt is denied or that the interface exposes only non-sensitive key identifiers, references, status, or metadata.

Test 2: Key Revocation Behavior

1. Configure the TOE to use an external customer-controlled key or key authorization mechanism.
2. Access protected database data as an authorized tenant user or tenant administrator.
3. Revoke, disable, or deny the TOE's authorization to use the external key according to the operational guidance.
4. Attempt to access protected database data in the TOE, repeating the attempt until the completed key-state enforcement interval of FCS_CKM_DBAAS_EXT.1.3 has elapsed.
5. Verify that access is denied no later than the expiration of the completed key-state enforcement interval, except for data available through an explicitly documented and evaluated recovery mechanism.
6. Verify that the transition to the key-unavailable enforcement state generates the audit record required by the module's auditable-events table, in the service event stream identified in the ST (Section 3.4, "Audit Channel Identification") — the record must be retrievable by the tenant while the database is not serving connections.
7. If a recovery mechanism is claimed, exercise it: verify that it cannot be completed without authorization by the tenant or an agent under the tenant's control, and that its use generates the audit record required by the module's auditable-events table in the ST-identified service event stream. Where the recovery mechanism is a restore from backup, this step may be performed jointly with FDP_BKP_EXT.1 Test 2, and the evaluator shall verify the restore tenant binding required there in the same run.

Continued access to protected data during the completed key-state enforcement interval is not a test failure; access continuing beyond its expiration is.

Test 2a: Key Provider Unreachability Behavior

Where the TSS documents a grace period during which the TOE continues operating on cached key material while the external key provider is unreachable, the evaluator shall, where the test environment permits interrupting the TOE's connectivity to the external key management service:

1. Access protected database data as an authorized tenant user or tenant administrator.
2. Make the external key management service unreachable to the TOE without revoking key authorization.
3. Verify that any continued access to protected data does not extend beyond the completed key-state enforcement interval, and that the TOE transitions to the key-unavailable enforcement state with the required audit record in the ST-identified service event stream.
4. Restore connectivity and verify that data access resumes only through the documented resumption behavior described in the TSS or operational guidance.

Where the test environment does not permit interrupting connectivity, the evaluator shall verify the unreachability behavior through provider-supported demonstration or provider test evidence, recorded in the ETR.

If the TOE caches key material, key handles, or derived keys after successful KMS authorization, the evaluator shall confirm during Test 2 and Test 2a that no cached material extends access to

protected data beyond the completed key-state enforcement interval.

Test 3: Key Material Handling Evidence

1. Review audit records, status views, error messages, diagnostic outputs, backup metadata, and management outputs that are tenant-accessible in the evaluated configuration.
2. Verify that plaintext customer-controlled master key material is not disclosed.
3. If provider-supported observation is used for non-tenant-accessible diagnostics, verify that the observation method and results are documented in the ETR.

Test 4: Customer Control of a KEM-Established Master Key

Applicable when customer-authorized ML-KEM establishment is selected in `FCS_CKM_DBAAS_EXT.1.1`.

1. Configure the evaluator-controlled external entity and cause the TOE to establish and derive the Master Key using the Crypto Module procedure.
2. Verify that the customer can authorize establishment and that protected database data is accessible after successful establishment.
3. Revoke, disable, or deny the customer's authorization, or make the customer-controlled establishment service unavailable, as described in the guidance.
4. Verify that the TOE prevents access to the protected database data no later than the assigned key-state enforcement interval, including expiration of any documented cache or reauthorization interval.
5. Verify through tenant-visible status or service-event-stream evidence that the failed or revoked authorization and key-unavailable enforcement transition are attributable to the affected customer or service instance without exposing the raw ML-KEM shared secret or derived Master Key.

Expected result: The KEM-established model remains customer-controlled after establishment; loss or revocation of customer authorization is enforced within the claimed bound, and neither KEM nor Master Key secret material is disclosed through DBaaS interfaces.

Chapter 6. FDP Class: User Data Protection

6.1. FDP_RIP.1/DBaaS Residual Information Protection

6.1.1. Evaluation Activities

6.1.1.1. TSS Activities

The evaluator shall review the TSS to determine whether tenant data can be present in shared DBMS memory, buffer cache, temporary storage, query work areas, diagnostic structures, or other shared DBMS resources. The evaluator shall verify that the TSS describes how such resources are cleared, reinitialized, or access-controlled before allocation to another tenant.

The evaluator shall verify that the TSS identifies which residual information protections are implemented by the TOE and which are provided by the TOE Platform, Trusted Platform, or operational environment.

6.1.1.2. Guidance Activities

The evaluator shall verify the operational guidance identifies any tenant-visible operations that release shared resources (for example, dropping temporary objects, closing sessions, or purging objects) and any configuration relevant to residual information protection in the evaluated configuration.

6.1.1.3. Test Activities

Test 1: Tenant-Interface Residual Data Test

1. Provision two tenant contexts in the evaluated configuration.
2. As Tenant A, create data with a unique pattern unlikely to appear elsewhere in the system.
3. Exercise TOE functionality that places the data into the claimed shared resource, such as queries that use buffer cache, temporary storage, sort areas, or query work areas.
4. Release the resource using documented tenant operations, such as dropping temporary objects, closing sessions, purging objects, or completing the workload.
5. As Tenant B, use documented tenant interfaces to allocate comparable resources and attempt to observe Tenant A’s unique pattern.
6. Verify that Tenant B cannot observe Tenant A’s data through tenant-accessible interfaces.

Where tenant-accessible interfaces permit repeatable testing, the evaluator shall perform the test through those interfaces. The evaluator is not required to inspect raw process memory unless such access is available in the evaluated configuration and permitted by the test environment.

6.2. FDP_ACC.1/DBaaS Tenant Isolation Policy

6.2.1. Evaluation Activities

6.2.1.1. TSS Activities

The evaluator shall verify the TSS identifies the subjects, objects, and operations covered by the Tenant Isolation SFP. The evaluator shall verify that the TSS identifies the tenant boundary used by the TOE, such as tenants, databases, pluggable databases, schemas, namespaces, accounts, tablespaces, tenant identifiers, or other DBMS constructs.

The evaluator shall also verify that the TSS scopes the Tenant Isolation SFP as subset access control consistent with the module's application note: tenant-scoped subjects, objects, and operations under this SFP, with subjects and objects outside its scope governed by the Base PP's access control SFPs in the composed ST.

6.2.1.2. Test Activities

The test activities for `FDP_ACC.1/DBaaS` are performed with the `FDP_ACF.1/DBaaS` tenant isolation policy rule tests.

6.3. FDP_ACF.1/DBaaS Tenant Isolation Policy Rules

6.3.1. Evaluation Activities

6.3.1.1. TSS Activities

The evaluator shall verify the TSS describes:

- the security attributes used to enforce the Tenant Isolation SFP, including tenant identifier, user identifier, service role, object owner, object tenant identifier, object type, and operation;
- how tenant subjects are bound to tenant identifiers;
- how tenant data objects, tenant metadata objects, and tenant audit records are labeled, scoped, or otherwise associated with tenant identifiers;
- the treatment of global system metadata visible to multiple tenants;
- any evaluated cross-tenant sharing mechanism and the authorization model for that mechanism; and
- how service provider administrative subjects are restricted from tenant data access except through interfaces and roles explicitly identified in the ST.

6.3.1.2. Guidance Activities

The evaluator shall verify the operational guidance describes how tenant boundaries are provisioned, how tenant users and service accounts are associated with tenant identifiers, and how any evaluated cross-tenant sharing mechanism is configured and revoked.

6.3.1.3. Test Activities

Test 1: Data Cross-Access Prevention

1. Provision Tenant A and Tenant B in the evaluated configuration.
2. Create tenant data objects in Tenant A with names and values known to the evaluator.
3. As Tenant B, attempt to access Tenant A's objects through documented SQL, API, management, import/export, backup, restore, and tenant administration interfaces available in the evaluated configuration.
4. Verify that all unauthorized attempts fail.

Test 2: Metadata Isolation

1. As Tenant A, create objects with distinct names.
2. As Tenant B, query system catalogs, metadata views, information schema views, management APIs, search interfaces, and monitoring views available to tenants.
3. Verify that Tenant B sees only Tenant B metadata and metadata explicitly defined as global system metadata.

Test 3: Tenant Audit Record Isolation

1. Generate audit events in Tenant A and Tenant B.
2. As Tenant A, query tenant-accessible audit records.
3. Verify that only Tenant A audit records are returned unless an evaluated provider, cross-tenant, or compliance function explicitly authorizes broader access.
4. Verify that Tenant B-specific usernames, object names, query text, or tenant data are not visible to Tenant A through tenant audit interfaces.

Test 4: Evaluated Cross-Tenant Sharing

If the ST claims an evaluated cross-tenant sharing mechanism:

1. Configure sharing between Tenant A and Tenant B according to the operational guidance.
2. Verify that Tenant B can access only the shared objects, metadata, or records authorized by Tenant A or by the evaluated sharing policy.
3. Revoke the sharing authorization.
4. Verify that Tenant B can no longer access the previously shared objects, metadata, or records.

6.4. FDP_BKP_EXT.1 Protected Backup and Restore

6.4.1. Evaluation Activities

6.4.1.1. TSS Activities

The evaluator shall verify the TSS describes:

- the backup and restore architecture for protected tenant data, including whether backups are created, stored, and restored by the TOE, the TOE Platform, or an external backup or object storage service;

- the reliance on FDP_DAR_EXT.1 of the DBMS Cryptographic Functions Module for the data-at-rest protection of backup data (this SD does not retest data-at-rest protection, which is covered by the DBMS Cryptographic Functions Module SD);
- how the TOE prevents a service provider administrator from restoring or exporting protected tenant data in plaintext without authorized use of the applicable customer-controlled master key; and
- how restored tenant data is bound to the originating tenant identifier so that a restore does not place data outside the Tenant Isolation SFP.

If backup storage or restore orchestration is provided by an external service, the evaluator shall verify the TSS identifies whether the restore authorization and tenant binding are TOE functionality or are addressed by OE.EXTERNAL_SERVICES; the data-at-rest protection of externally stored backups is assessed under FDP_DAR_EXT.1.

6.4.1.2. Guidance Activities

The evaluator shall verify the operational guidance describes how backups are configured, how restore is authorized, the roles permitted to perform restore, and any customer-visible status or audit information for backup and restore operations.

6.4.1.3. Test Activities

Note: The data-at-rest confidentiality of backup data is evaluated under FDP_DAR_EXT.1 of the DBMS Cryptographic Functions Module SD and is not retested here. The activities below address the backup and restore-path properties specific to this module.

Test 1: Provider Plaintext Restore Prevention

1. As a Service Provider Administrator role, or the closest role available in the evaluated configuration, attempt to restore or export Tenant A protected data in plaintext without authorized use of the customer-controlled key. Where no such role is available to the evaluator, this test shall be performed as a provider-supported demonstration under the Evaluator Access Strategy (Section 3.3, “Evaluator Access Strategy”), with the demonstration method and results recorded in the ETR.
2. Verify that the operation is denied, or that any restored or exported artifact remains ciphertext under the customer-controlled key hierarchy. **Pass criterion:** absent an authorized grant of the applicable customer-controlled key, no step of the restore or export path yields plaintext Tenant A data.

Test 2: Restore Tenant Binding

1. Restore a Tenant A backup according to the operational guidance.
2. As Tenant B, attempt to access the restored Tenant A data.
3. Verify that restored data remains within the Tenant A isolation boundary and is not accessible to Tenant B.

Chapter 7. FRU Class: Resource Utilisation

7.1. FRU_RSA.1/DBaaS and FRU_RSG_EXT.1 Resource Governance

The activities below address the quota enforcement required by `FRU_RSA.1/DBaaS` and the enforcement action required by `FRU_RSG_EXT.1` together, as they are exercised by the same workloads.

7.1.1. Evaluation Activities

7.1.1.1. TSS Activities

The evaluator shall verify the TSS describes:

- the DBMS-managed resources subject to tenant quotas or limits (the completion of the FRU_RSA.1.1/DBaaS selection);
- the resource governance mechanism, such as resource managers, quotas, workload queues, throttling, session limits, or query governors;
- the enforcement action claimed in FRU_RSG_EXT.1.1 for operations that exceed configured limits;
- the authorized roles that can configure resource governance; and
- how resource governance is applied to tenants, users, service accounts, and workloads.

7.1.1.2. Guidance Activities

The evaluator shall verify the operational guidance describes how to configure tenant resource limits and how to monitor quota enforcement.

7.1.1.3. Test Activities

Test 1: Resource Limit Enforcement

1. Configure a resource quota for a tenant according to the operational guidance.
2. Initiate a workload that exceeds the configured limit.
3. Verify that the TOE performs the enforcement action claimed in FRU_RSG_EXT.1.1, such as queueing, throttling, rejection, or termination.
4. Verify that the TOE remains available for other tenant operations that do not exceed their configured limits.

Test 2: Unauthorized Resource Governance Modification

1. Authenticate as a role that is not authorized to modify resource governance settings.
2. Attempt to increase or disable the tenant resource quota.

3. Verify that the operation is denied.

Chapter 8. FIA Class: Identification and Authentication

8.1. FIA_USB.1/DBaaS User-Subject Binding

8.1.1. Evaluation Activities

8.1.1.1. TSS Activities

The evaluator shall verify the TSS describes how tenant identifier, user identifier, and service role attributes are associated with subjects acting on behalf of users or service accounts, and how the TOE prevents tenant-controlled inputs from changing the bound tenant identifier except through an evaluated mechanism. Where automated autonomous management logic is selected in `FMT_SMR.1/DBaaS`, the evaluator shall verify the TSS describes how subjects acting on its behalf within a tenant context bind the affected tenant's identifier, as required by the module's application note.

8.1.1.2. Guidance Activities

The evaluator shall verify the operational guidance describes how tenant users and service accounts are provisioned so that they bind to the correct tenant identifier, and identifies any evaluated mechanism by which a bound tenant identifier can legitimately change.

8.1.1.3. Test Activities

Test 1: Tenant ID Binding Verification

1. Authenticate as a tenant user.
2. Execute a documented query or API call that returns the current tenant context, session attributes, or equivalent DBMS context.
3. Verify that the returned tenant identifier matches the authenticated tenant.
4. Attempt to change the tenant identifier by using user-controlled session parameters, connection attributes, request headers, or SQL variables that are available to the tenant.
5. Verify that unauthorized tenant identifier changes are denied or ignored.

Test 2: Audit Correlation

1. Perform an auditable action as Tenant A.
2. Query the audit trail through tenant-accessible interfaces.
3. Verify that the audit record is stamped with the correct tenant identifier and user identifier.
4. Authenticate as Tenant B.
5. Attempt to access the Tenant A audit record.
6. Verify that Tenant B cannot access the Tenant A audit record unless an evaluated sharing or compliance function explicitly permits access.

Chapter 9. FMT Class: Security Management

9.1. FMT_SMR.1/DBaaS Security Roles

9.1.1. Evaluation Activities

9.1.1.1. TSS Activities

The evaluator shall verify the TSS identifies the roles maintained by the TOE and provides the mapping required by the module's FMT_SMR.1/DBaaS application note from each module role label — Tenant User, Tenant Administrator, Service Provider Administrator — to the TOE role, privilege set, group, or account class that realizes it, together with any automated autonomous management logic or other roles claimed in the ST. The TOE is not required to name its roles verbatim; the evaluator shall instead verify that every label is mapped, that the mapping is unambiguous, and that no single TOE role combines Service Provider Administrator authority with tenant-scoped authority in a way that vacates the separation enforced by the `FMT_MOF.1/DBaaS` iterations, `FDP_ACF.1/DBaaS`, or `FCS_CKM_DBAAS_EXT.1`. The evaluator shall verify the TSS describes how users and service accounts are associated with these roles and how the roles map to the management restrictions in the `FMT_MOF.1/DBaaS` iterations and the access-control rules in `FDP_ACF.1/DBaaS`.

9.1.1.2. Guidance Activities

The evaluator shall verify the operational guidance describes how each role is assigned, the privileges associated with each role, and the separation between Tenant Administrator and Service Provider Administrator capabilities.

9.1.1.3. Test Activities

Test 1: Role Maintenance and Association

1. Using the interfaces available in the evaluated configuration, enumerate the roles recognized by the TOE.
2. Verify that the TOE roles to which the ST maps the module's role labels are present under their TOE names.
3. Associate a test user with a tenant role and verify that the user receives the privileges defined for that role and no others.
4. For roles that are not tenant-visible (Service Provider Administrator, automated autonomous management logic), verify their presence and privilege boundaries through a provider-supported demonstration or provider role documentation under the Evaluator Access Strategy (Section 3.3, "Evaluator Access Strategy"), recorded in the ETR.

9.2. FMT_SMF.1/DBaaS Specification of Management Functions

9.2.1. Evaluation Activities

9.2.1.1. TSS Activities

The evaluator shall verify the TSS identifies the management functions provided by the TOE, including configuration of customer-controlled key references and key-management integration, tenant resource quotas, audit export and retention, update deployment (including tenant deferral policy or maintenance windows where selected), and any evaluated cross-tenant sharing mechanism. The evaluator shall verify the TSS identifies, for each management function, the authorized roles defined by FMT_SMR.1/DBaaS.

9.2.1.2. Guidance Activities

The evaluator shall verify the operational guidance describes how to perform each claimed management function and which role is authorized to perform it.

9.2.1.3. Test Activities

Test 1: Management Function Availability

1. For each claimed management function, perform the function as an authorized role using the interfaces available in the evaluated configuration.
2. Verify that the function operates as described.
3. Attempt the same function as an unauthorized role and verify that it is denied. This test may be performed jointly with the FMT_MOF.1/DBaaS iteration tests.
4. For management functions restricted to provider roles that are not available to the evaluator, perform the authorized-role leg through a provider-supported demonstration under the Evaluator Access Strategy (Section 3.3, “Evaluator Access Strategy”), recorded in the ETR; the unauthorized-role denial leg shall still be tested directly through tenant interfaces.
5. Where automated autonomous management logic is selected in FMT_SMR.1/DBaaS, identify at least one record of an autonomous management action in the ST-identified audit channel and verify that it identifies the autonomous subject as the actor and, where the action affected a tenant context, carries the affected tenant’s identifier and is visible to that tenant.

9.3. FMT_MOF.1/DBaaS-Disable and FMT_MOF.1/DBaaS-Modify Management of Security Functions Behaviour

9.3.1. Evaluation Activities

9.3.1.1. TSS Activities

The evaluator shall verify the TSS identifies the security functions that cannot be disabled in the evaluated configuration (FMT_MOF.1/DBaaS-Disable) and the roles authorized to modify DBaaS security functions such as resource governance, patching and updates, and audit export (FMT_MOF.1/DBaaS-Modify).

9.3.1.2. Guidance Activities

The evaluator shall verify the operational guidance describes the roles authorized to modify each claimed DBaaS security function and the expected behavior when unauthorized roles attempt modification.

9.3.1.3. Test Activities

Test 1: Audit Immutability Control

1. Authenticate as a tenant administrator.
2. Attempt to disable audit generation for TOE-required audit events.
3. Verify that the operation is denied.

Test 2: Encryption Immutability Control

1. Authenticate as a tenant administrator.
2. Attempt to disable data-at-rest encryption for protected tenant data.
3. Verify that the operation is denied.

Test 3: Role Restriction Verification

1. Authenticate as a tenant administrator or other role not authorized to manage provider or root-level functions.
2. Attempt to grant provider, root, or equivalent administrative privileges to a standard tenant user.
3. Verify that the operation is denied.

Chapter 10. FAU Class: Security Audit

10.1. FAU_SEL.1/DBaaS Selective Audit

10.1.1. Evaluation Activities

10.1.1.1. TSS Activities

The evaluator shall verify the TSS describes how audit event selection is scoped by tenant identifier, user identifier, service role, object identifier, and event type. Restriction of audit record queries to the owning tenant is an access-control rule of `FDP_ACF.1/DBaaS` and is evaluated there; Test 1 below may be performed jointly with the `FDP_ACF.1/DBaaS` tests.

10.1.1.2. Guidance Activities

The evaluator shall verify the operational guidance describes how audit selection is configured, the attributes on which selection can be based, and the roles authorized to change tenant-scoped audit selection.

10.1.1.3. Test Activities

Test 1: Audit Scope Leakage

1. Generate audit events in Tenant A and Tenant B.
2. As Tenant A administrator, query the audit logs.
3. Verify that only Tenant A records are returned unless an evaluated provider, cross-tenant, or compliance function explicitly authorizes broader access.
4. Analyze the query results to ensure no Tenant B data, usernames, object names, query text, or tenant identifiers are visible to Tenant A unless explicitly authorized by the evaluated configuration.

10.2. FAU_STG.2/DBaaS Protected Audit Trail Storage (Immutable Audit Stream)

10.2.1. Evaluation Activities

10.2.1.1. TSS Activities

The evaluator shall verify the TSS describes how the TOE prevents unauthorised modification and deletion of stored audit records, including that no tenant-facing interface provides audit modification or deletion functions. If audit records are exported to an external sink, the evaluator shall verify the TSS identifies whether the export mechanism, external sink, or both are part of the TOE or operational environment.

`FAU_STG.2/DBaaS` applies to the DBMS audit trail. Protection, tenant visibility, and retrieval of service event stream records are defined by the module's auditable-events refinement and verified under

Section 3.4, “Audit Channel Identification”; they are not retested here.

10.2.1.2. Guidance Activities

The evaluator shall verify the operational guidance describes audit retention, audit export configuration, authorized audit query roles, and any recovery or archival mechanism for audit records.

10.2.1.3. Test Activities

Test 1: Deletion and Modification Prevention

1. As a tenant administrator, identify an audit record generated by the TOE.
2. Attempt to delete or modify that record through documented tenant interfaces, such as SQL DELETE or UPDATE commands, management APIs, or tenant audit tools.
3. Verify that the operation is denied.

Test 2: Audit Export Verification

1. Configure audit export to an external sink according to the operational guidance. Where audit export configuration is restricted to provider roles not available to the evaluator, this step shall be performed through a provider-supported demonstration under the Evaluator Access Strategy (Section 3.3, “Evaluator Access Strategy”), recorded in the ETR.
2. Generate audit events.
3. Verify that the external sink receives the expected events.
4. Attempt to stop, redirect, or weaken the audit export pipeline as a tenant administrator.
5. Verify that the unauthorized operation is denied.

Chapter 11. FPT Class: Protection of the TSF

11.1. FPT_TUD_DBAAS_EXT.1 Trusted Service Update

11.1.1. Evaluation Activities

11.1.1.1. TSS Activities

The evaluator shall verify the TSS describes:

- the end-to-end update process for the DBMS service software, from the provider's authorized update process through deployment, including the deployment model selected in FPT_TUD_DBAAS_EXT.1.1 (uniform rolling deployment, tenant-configurable deferral, or tenant-selectable maintenance windows);
- the deployment artifacts to which signature verification applies, the signature format used (for example X.509 certificate-based, OpenPGP, or detached signatures over deployment artifacts), and the point in the deployment path at which verification occurs — before the updated software is placed into service;
- the signature algorithm and hash, as completions of the `FCS_COP.1/SigVer` component included through the DBMS Cryptographic Functions Module, and how the verification key is integrity protected;
- which parts of the update pipeline are TSF and which reside in the operational environment, together with the TOE-side enforcement points (an ST in which update handling lies wholly outside the TOE boundary cannot claim this component); and
- the mechanism by which tenants determine the version or revision identifier of the service software currently serving them, and how that identifier maps to deployed service software.

Where certificate path validation is part of signature verification, the evaluator shall verify the TSS identifies the applicable components of the Functional Package for X.509 Certificates [[X509_FP]].

11.1.1.2. Guidance Activities

The evaluator shall verify the operational guidance describes:

- how a tenant queries the current service version or revision identifier;
- customer-visible update status and notifications;
- update deferral options or maintenance windows, where those selections are claimed, including the maximum deferral period; and
- any tenant-visible configuration related to updates.

11.1.1.3. Test Activities

Note: Testing live patching in a production DBaaS environment can be disruptive, and update deployment is a provider-controlled operation. Where the evaluator cannot initiate or observe an update directly, the evaluator shall use the Evaluator Access Strategy (Section 3.3, "Evaluator Access

Strategy”): a provider-supported demonstration of the update path on a non-production instance, together with audit records and deployment evidence, recorded in the ETR.

Test 1: Update Verification and Provenance

1. Through a provider-supported demonstration or on a non-production instance, observe the deployment of a service software update.
2. Verify that the deployment evidence identifies the update source as the provider’s authorized update process.
3. Verify that signature verification of the identified deployment artifacts occurs before the updated software is placed into service, using the algorithm and key documented in the TSS.
4. Where feasible in the demonstration environment, present an artifact with an invalid or missing signature and verify that it is not placed into service and that the failure is audited.

Test 2: Tenant Non-Interference

1. Authenticate as a tenant administrator.
2. Attempt to block, bypass, or indefinitely delay security updates using tenant-accessible configuration settings, beyond any deferral or maintenance-window allowance claimed in FPT_TUD_DBAAS_EXT.1.1.
3. Verify that updates are applied according to the evaluated update policy, and that any claimed deferral is bounded by the documented maximum deferral period.

Test 3: Tenant Version Visibility

1. As a tenant user or tenant administrator, query the service version or revision identifier through the documented interface.
2. Verify that an identifier is returned and matches the deployed service software identified in the update evidence of Test 1 or in provider-supported deployment records.
3. Where an update is observed during the evaluation (Test 1), verify that the identifier changes accordingly.

Appendix A: Cross-Reference to CEM Work Units

The Evaluation Activities of this SD are narrative refinements of CEM:2022 work units: TSS Activities refine the ASE_TSS.1 examination work units, Guidance Activities refine the AGD_OPE.1 and AGD_PRE.1 examination work units, and Test Activities refine the ATE_IND.2 test-devising and test-conduct work units. The table below enumerates this mapping for every SFR section of this SD, followed by the mapping for the DBaaS-specific SAR considerations of the Assurance Activities Overview. Provider-supported demonstrations performed under the Evaluator Access Strategy are conducted and recorded as part of the ATE_IND.2 (or, for AVA activities, AVA_VAN.2) work units they support.

A.1. SFR Evaluation Activities to CEM Mapping

SFR Section	CEM Work Unit(s)	Notes
FCS_CKM_DBAAS_EXT.1	ASE_TSS.1-1; AGD_OPE.1-1, AGD_OPE.1-4; ATE_IND.2-3, ATE_IND.2-4	Tests 1–4, including provider-inaccessibility, revocation, and conditional KEM-established customer-control testing; coordinates with the Crypto Module SD for key lifecycle, role-specific ML-KEM, derivation, and trusted channel
FDP_RIP.1/DBaaS	ASE_TSS.1-1; AGD_OPE.1-1; ATE_IND.2-3, ATE_IND.2-4	Tenant-interface residual data testing
FDP_ACC.1/DBaaS	ASE_TSS.1-1	Test coverage provided under FDP_ACF.1/DBaaS
FDP_ACF.1/DBaaS	ASE_TSS.1-1; AGD_OPE.1-1, AGD_OPE.1-4; ATE_IND.2-3, ATE_IND.2-4	Tests 1–4 (data, metadata, audit isolation, and evaluated sharing)
FDP_BKP_EXT.1	ASE_TSS.1-1; AGD_OPE.1-1, AGD_OPE.1-4; ATE_IND.2-3, ATE_IND.2-4	Test 1 may be performed as a provider-supported demonstration; data-at-rest confidentiality of backups is evaluated under the Crypto Module SD
FRU_RSA.1/DBaaS and FRU_RSG_EXT.1	ASE_TSS.1-1; AGD_OPE.1-1; ATE_IND.2-3, ATE_IND.2-4	Joint quota-enforcement and enforcement-action testing
FIA_USB.1/DBaaS	ASE_TSS.1-1; AGD_OPE.1-1; ATE_IND.2-3, ATE_IND.2-4	Tenant-binding and audit-correlation testing
FMT_SMR.1/DBaaS	ASE_TSS.1-1; AGD_OPE.1-1, AGD_OPE.1-4; ATE_IND.2-3, ATE_IND.2-4	Provider-role verification may use provider-supported demonstration

SFR Section	CEM Work Unit(s)	Notes
FMT_SMF.1/DBaaS	ASE_TSS.1-1; AGD_OPE.1-1, AGD_OPE.1-4; ATE_IND.2-3, ATE_IND.2-4	Provider-restricted functions may use provider-supported demonstration for the authorized leg
FMT_MOF.1/DBaaS-Disable and FMT_MOF.1/DBaaS-Modify	ASE_TSS.1-1; AGD_OPE.1-1, AGD_OPE.1-4; ATE_IND.2-3, ATE_IND.2-4	Tests 1–3 (audit and encryption immutability, role restriction)
FAU_SEL.1/DBaaS	ASE_TSS.1-1; AGD_OPE.1-1; ATE_IND.2-3, ATE_IND.2-4	Audit-scope leakage testing, jointly with FDP_ACF.1/DBaaS
FAU_STG.2/DBaaS	ASE_TSS.1-1; AGD_OPE.1-1, AGD_OPE.1-4; ATE_IND.2-3, ATE_IND.2-4	Export configuration step may use provider-supported demonstration
FPT_TUD_DBAAS_EXT.1	ASE_TSS.1-1; AGD_OPE.1-1, AGD_OPE.1-4; ATE_IND.2-3, ATE_IND.2-4	Update verification typically performed as a provider-supported demonstration on a non-production instance at the evaluated service software revision

A.2. SAR Considerations to CEM Mapping

SD Section	CEM Work Unit(s)	Notes
ASE: Security Target Evaluation	ASE_INT.1, ASE_CCL.1, ASE_SPD.1, ASE_OBJ.2, ASE_TSS.1	Supplements the Base PP ASE activities with managed-service verification
ADV: Development	ADV_ARC.1, ADV_FSP.2, ADV_TDS.1	Managed-service interface scope and tenant-separation architecture
ALC: Life-cycle Support	ALC_CMC.2, ALC_CMS.2, ALC_DEL.1, ALC_FLR.3	Continuous-deployment CM identification, deployment-path delivery interpretation, and flaw-remediation coupling to FPT_TUD_DBAAS_EXT.1
ATE: Tests	ATE_COV.1, ATE_FUN.1, ATE_IND.2	Tenant-vantage independent testing, provider test evidence for provider-restricted functions, revision-parity requirement for non-production instances
AVA: Vulnerability Assessment	AVA_VAN.2	Tenant-vantage vulnerability analysis and penetration testing with production-safety constraints
Documentation Requirements (AGD)	AGD_OPE.1, AGD_PRE.1	DBaaS-specific operational guidance and service-provisioning preparative procedures

Appendix B: Evidence Vantage Mapping

The table below assigns each test activity of this SD to the evidence vantages defined in Section 2.5, "Evidence Vantages". The assignment is informative: it states where evidence is **expected** to be collected for a typical provider-operated DBaaS TOE, so that laboratories can plan tenancy provisioning, external-service setup, and provider participation when scoping the evaluation. The evaluated TOE's interface exposure governs — the ST's interface identification and role-label mapping determine which functions sit at V2 rather than V4 for a given TOE, and the evaluator records deviations from this table in the ETR under the Evaluator Access Strategy.

TSS and Guidance Activities are documentation review and are not listed; only Test Activities appear.

Table 2. Evidence vantages by test activity

Test Activity	Vantage(s)	Evidence collected
FCS_CKM_DBAAS_EXT.1 Test 1 (Provider Inaccessibility)	V1, V2	Key-export and key-display attempts through every tenant-facing database and management interface; denial or metadata-only responses.
FCS_CKM_DBAAS_EXT.1 Test 2 (Key Revocation)	V3 → V1, V2	Revocation induced at the evaluator's external KMS (V3); denial of data access observed at the database interfaces (V1); enforcement-state transition and its record verified in the ST-identified service event stream, retrieved through its tenant interface (V2).
FCS_CKM_DBAAS_EXT.1 Test 2a (Key Provider Unreachability)	V3 → V1, V2; V4 fallback	KMS connectivity interrupted (V3); grace-period behavior, enforcement-state transition, and resumption observed at V1/V2. Where connectivity cannot be interrupted, provider-supported demonstration or provider test evidence (V4).
FCS_CKM_DBAAS_EXT.1 Test 3 (Key Material Handling Evidence)	V1, V2; V4 for non-tenant diagnostics	Tenant-accessible audit records, status views, error messages, and backup metadata reviewed for key-material disclosure; provider-supported observation for diagnostics not exposed to tenants.
FCS_CKM_DBAAS_EXT.1 Test 4 (KEM-Established Customer Control)	V3 → V1, V2	ML-KEM peer and authorization controlled at V3; successful establishment and data use observed at V1/V2; revocation or unavailability induced at V3; bounded denial and the service-event-stream transition verified at V1/V2.

Test Activity	Vantage(s)	Evidence collected
FDP_RIP.1/DBaaS Test 1 (Residual Data)	V1 (V2 to provision)	Two tenant contexts provisioned at V2; unique-pattern write, resource release, and cross-tenant observation attempts all at V1.
FDP_ACF.1/DBaaS Tests 1–4 (Tenant Isolation)	V1 (V2 to provision and configure sharing)	Cross-tenant data, metadata, and audit access attempts at V1; tenant provisioning and any evaluated sharing configuration at V2.
FDP_BKP_EXT.1 Test 1 (Provider Plaintext Restore Prevention)	V4 (V2 where a closest-available role exists)	Provider-supported demonstration of a plaintext restore/export attempt without the customer-controlled key; ciphertext pass criterion verified in the demonstration output.
FDP_BKP_EXT.1 Test 2 (Restore Tenant Binding)	V2 → V1	Restore initiated from the tenant service control plane; tenant-binding of restored data verified through cross-tenant access attempts at V1.
FRU_RSA.1/DBaaS and FRU_RSG_EXT.1 Test 1 (Resource Limit Enforcement)	V2 → V1	Quota configured at V2; over-limit workload driven and enforcement action observed at V1.
FRU_RSA.1/DBaaS and FRU_RSG_EXT.1 Test 2 (Unauthorized Quota Modification)	V1, V2	Quota-modification attempt by an unauthorized role, at whichever plane exposes the setting; denial observed.
FIA_USB.1/DBaaS Tests 1–2 (Tenant Binding, Audit Correlation)	V1	Tenant-context queries, binding-manipulation attempts, and audit-record correlation through database interfaces.
FMT_SMR.1/DBaaS Test 1 (Role Maintenance)	V1, V2; V4 for provider roles	Enumeration of mapped TOE roles and privilege verification at tenant vantages; provider-role presence and boundaries by provider-supported demonstration or provider role documentation.
FMT_SMF.1/DBaaS Test 1 (Management Functions)	V2 (V1 for database-level functions); V4 for the authorized leg of provider-restricted functions	Each claimed management function exercised as an authorized role; denial leg always tested directly from tenant vantages.
FMT_MOF.1/DBaaS Tests 1–3 (Immutability and Role Restriction)	V1, V2	Tenant-administrator attempts to disable audit, disable encryption, and escalate privileges; denial observed.
FAU_SEL.1/DBaaS Test 1 (Audit Scope Leakage)	V1	Cross-tenant audit queries and result analysis through tenant audit interfaces.

Test Activity	Vantage(s)	Evidence collected
FAU_STG.2/DBaaS Test 1 (Deletion/Modification Prevention)	V1	Audit-record tamper attempts through documented tenant interfaces; denial observed.
FAU_STG.2/DBaaS Test 2 (Audit Export)	V2 (V4 where provider-restricted) → V3; tamper leg V1, V2	Export configured at V2 or by provider-supported demonstration; delivery verified at the evaluator's external sink (V3); pipeline-tamper attempts from tenant vantages.
FPT_TUD_DBAAS_EXT.1 Test 1 (Update Verification and Provenance)	V4	Provider-supported demonstration of the update path on a revision-parity non-production instance, including signature verification and, where feasible, an invalid-signature artifact.
FPT_TUD_DBAAS_EXT.1 Test 2 (Tenant Non-Interference)	V2	Tenant-accessible update-configuration settings exercised against the evaluated update policy and deferral bound.
FPT_TUD_DBAAS_EXT.1 Test 3 (Tenant Version Visibility)	V1, V2	Service revision identifier queried through the documented tenant interface and matched against deployment evidence.

Laboratories scoping an evaluation against the DBaaS PP-Configuration should note the V4 concentrations: FDP_BKP_EXT.1 Test 1 and FPT_TUD_DBAAS_EXT.1 Test 1 require provider participation in substantially all evaluations, and FMT_SMR.1/DBaaS, FMT_SMF.1/DBaaS, FAU_STG.2/DBaaS Test 2, and the FCS_CKM_DBAAS_EXT.1 Test 2a fallback require it whenever the evaluated TOE restricts the relevant function to provider roles. All remaining test activities are executable from the evaluation tenancy and evaluator-controlled external services without provider involvement.

Appendix C: Test Environment Guidance for DBaaS Evaluators

C.1. Recommended Test Configuration

The following configuration is recommended for independent testing of a provider-operated DBaaS TOE:

- **Tenant contexts**: at least two evaluator-provisioned tenant contexts (Tenant A and Tenant B) in the evaluated configuration; a third tenant context is recommended where an evaluated cross-tenant sharing mechanism is claimed, so that sharing and non-sharing tenants can be distinguished;
- **Customer-controlled keys**: an external key management service instance under evaluator control, holding a revocable test key, so that FCS_CKM_DBAAS_EXT.1 revocation behavior can be exercised without provider involvement; when ML-KEM establishment is selected, the service or a complementary test harness also supports the claimed parameter set and TOE role without exposing shared-secret material through operational interfaces;
- **Audit sink**: an evaluator-controlled external audit sink for FAU_STG.2/DBaaS export testing;
- **Workload tooling**: load-generation tooling sufficient to exceed configured tenant resource quotas for the FRU tests;
- **Demonstration environment**: for provider-supported demonstrations (update path, provider-role functions, restore testing), a non-production instance of the service verified — via the FPT_TUD_DBAAS_EXT.1.3 identifier or CM evidence — to run the same service software revision as the evaluated configuration.

C.2. Production-Safety Constraints

Because the TOE is a live, shared, provider-operated service:

- Isolation and resource-exhaustion testing shall be directed only at evaluator-provisioned tenant contexts; the evaluator shall never target tenants not provisioned for the evaluation;
- Resource-exhaustion workloads shall be coordinated with the provider and bounded so that enforcement, not platform degradation, is what the test observes;
- Update-path and restore testing shall be performed on the non-production demonstration instance unless the provider supports safe execution in the evaluated configuration;
- The evaluator shall respect the service terms applicable to the evaluation tenancy and record any provider-imposed constraint that limits a test, together with the compensating evidence used, in the ETR.

C.3. Test Data Handling

- Use synthetic test data containing no actual sensitive information; unique, searchable patterns (as required by the FDP_RIP.1/DBaaS test) shall be generated for the evaluation;

- Delete evaluation tenants, test data, keys, and audit sinks after the evaluation completes;
- Document data handling, tenant deprovisioning, and key destruction in the ETR.

Appendix D: Document References

- [CC1] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model, CCMB-2022-11-001, CC:2022 Revision 1, November 2022.
- [CC2] Common Criteria for Information Technology Security Evaluation, Part 2: Security functional requirements, CCMB-2022-11-002, CC:2022 Revision 1, November 2022.
- [CC3] Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance requirements, CCMB-2022-11-003, CC:2022 Revision 1, November 2022.
- [CC4] Common Criteria for Information Technology Security Evaluation, Part 4: Framework for the specification of evaluation methods and activities, CCMB-2022-11-004, CC:2022 Revision 1, November 2022.
- [CC5] Common Criteria for Information Technology Security Evaluation, Part 5: Pre-defined packages of security requirements, CCMB-2022-11-005, CC:2022 Revision 1, November 2022.
- [CCE] Common Criteria for Information Technology Security Evaluation, Errata and interpretation for CC:2022 (Release 1) and CEM:2022 (Release 1), CCMB-002, Version 1.1, July 22, 2024.
- [CEM] Common Methodology for Information Technology Security Evaluation, Evaluation methodology, CCMB-2022-11-006, CEM:2022 Revision 1, November 2022.
- [DBMS_MOD_DBAAS] collaborative PP-Module for Database-as-a-Service (DBaaS), Version 0.5, 2026-07-20.
- [cPP_DBMS] collaborative Protection Profile for Database Management Systems, Version 2.0, 27 April 2026.
- [cPP_DBMS_SD] Supporting Document Mandatory Technical Document Evaluation Activities for the collaborative Protection Profile for Database Management Systems, Version 2.0, 27 April 2026.
- [DBMS_MOD_CRYPTO] collaborative PP-Module for DBMS Cryptographic Functions, Version 0.5, 2026-07-20.
- [DBMS_MOD_CRYPTO_SD] Supporting Document - Evaluation Activities for DBMS Cryptographic Functions Module, Version 0.5, 2026-07-20.
- [Crypto_Catalog] Specification of Functional Requirements for Cryptography (CCDB-018), Version 1.0, January 2025.
- [TLS_FP] Functional Package for Transport Layer Security (TLS), Version 2.1, 2025-08-25.
- [X509_FP] Functional Package for X.509 Certificates, Version 1.0. Available at commoncriteria.github.io/X509.
- [CCiTC_SaaS] Common Criteria in the Cloud Technical Community, Guidance for Cloud Evaluations, Version 1.1.