

Access risk analysis built for Oracle E-Business Suite

EBS grants access by responsibility and governs its reach through Multi-Org — a shape a generic duty-segregation engine tends to flatten. KeyForge models EBS the way EBS actually works, then runs that same engine across the rest of your estate and the identities that aren't people.

- Responsibility, function and menu granularity
- Operating-unit aware
- Cross-application segregation of duties
- People, service accounts and agents

THE EBS PROBLEM

Operating units decide whether a conflict is real

In EBS, access is granted by **responsibility**, and what a responsibility can reach is governed by the **security profile** attached to it — one operating unit, several, or global. Two conflicting responsibilities held against two different operating units are not a conflict. When that dimension is flattened away, the register fills with findings the user could never have acted on.

WITHOUT OPERATING-UNIT CONTEXT Responsibility treated as a plain role <i>AP Invoice Entry</i> in the US operating unit plus <i>AP Payments Manager</i> in the EMEA operating unit is reported as a critical payment-fraud conflict. The user can never act on both. The finding is a false positive, and it returns on every run.	THE KEYFORGE MODEL Operating-unit scope inside the rule Every rule declares how operating-unit scope is evaluated. A same-unit rule fires only when both sides of the conflict land in one operating unit, so the finding you hand an auditor is a finding the user could actually act on.
--	---

Same unit	Anywhere	Named pairs	Across units	Global access only
Both sides must fall in one operating unit for the conflict to count.	Scope is irrelevant — the combination is toxic wherever it occurs.	Only for enumerated combinations, such as US paired with EMEA.	Fires only when the two sides sit in different operating units.	Reserved for responsibilities carrying unrestricted org access.

Scope is not a report filter applied after analysis. It is evaluated inside the rule, so the violation count you present is the count that is actually true.

What the module does

Fine-grained risk analysis Analyse at three levels of grain: the responsibility itself, the business function, or the individual form function and menu entry beneath it. Coarse rules that keep the register readable and surgical rules that remove false positives run side by side in the same analysis.	Pre-built rule libraries Ships with an Oracle controls library of more than nine thousand rules spanning Financials, Human Capital, Procurement and Order-to-Cash, so a first meaningful risk picture arrives in days rather than quarters. Your own matrix loads directly from a spreadsheet, and any library can be exported for review or handover.
Mitigating control recommendations Controls are first-class records — preventive or detective, with a test frequency and a named owner — and are pre-linked to the rules they mitigate. When a violation surfaces, the recommended control set is already attached, so the reviewer assigns rather than researches. Assignments carry effective dates and lapse on schedule.	Segregation of duties and risk policy review A reviewer workqueue with ownership, bulk disposition, threaded discussion and a full evidence trail behind every finding: which user, which responsibility, which function, which operating unit, which rule. Role-pair review shows which combinations can conflict before anyone is assigned to them.
Inheritance lineage Every finding opens to the complete chain from user to function, and where access arrives through several responsibilities at once, all of the paths are shown. Revoking one responsibility often does not clear the violation — lineage names the grants that do, so remediation is surgical rather than removing access someone needs.	Audit-ready reporting Every analysis is a dated, durable record of its ruleset, its population and its results — reproducible months later for a walkthrough. Findings export with their evidence trail intact, rule libraries export as point-in-time snapshots, and the change history exports alongside them.
What-if analysis for role design Four modes of simulation: add or remove responsibilities for an existing user; model a new hire from an empty baseline; test a single responsibility for conflicts it carries on its own; and compare two responsibilities to separate intrinsic risk from risk that appears only in combination. Role-conflict results are precomputed, so redesign stays interactive across thousands of roles.	Compensating controls and exceptions Exceptions are requested with a written justification, approved or rejected by a separate party, and bounded by a validity window with its own recertification date. Nothing is permanent by default — expiry runs on schedule and the violation returns to the queue for a fresh decision.
Continuous monitoring of high-risk access Two enforcement points, not one. Scheduled analysis re-scores the whole population against the current rules; a request-time check evaluates the conflict <i>before</i> the responsibility is granted, so provisioning is blocked or escalated rather than remediated later. New high and critical findings are pushed to your monitoring, ticketing and alerting systems as they arise.	Service accounts, API keys and agents A non-human identity register that carries what audit actually asks for: named custodian, subject-matter expert and backup owner, environment, data classification, SOX scope, vault linkage, rotation interval, credential age, last-used date and just-in-time eligibility. Stale, unrotated, unowned and unvaulted accounts surface as standing measures — and machine entitlements are analysed by the same engine, so an integration account holding both invoice entry and payment release is a violation like any other.

Cross-application segregation of duties A single rule spans systems: EBS payment release against Oracle Cloud supplier creation, SAP vendor master against EBS invoice entry, a Workday worker record against an EBS payables function. Rules are shared across libraries, so a cross-application rule is authored once and enforced in every library that should see it, including the one checked at request time.	Multiple rule libraries, versioned Run as many libraries as the business needs — external audit, internal audit, a business-unit variant, a revision under review — at the same time and against the same population. A library clones in a single step to create the next version, revisions stay in draft until promoted, and shared rules mean a correction authored once propagates everywhere it belongs.
Approvals on risk register changes Exceptions and compensating control assignments move through request, review and decision with a named requester, a named approver, a written justification and a bounded validity period. Rule and library revisions are authored as drafts and promoted deliberately, so nothing reaches the live register by accident.	Complete history of register changes Every change to the risk register is captured with the full before and after state, the person who made it, the moment it happened and where it came from — rules, libraries, mitigating controls, exceptions, violation dispositions, organisational scopes, connected systems and reference data alike. An auditor asking who loosened a rule, when and from what gets a direct answer rather than a reconstruction.

WHERE KEYFORGE GOES FURTHER

The same rules, extended to the identities that aren't people

EBS integration users, concurrent-program owners and shared administrative logins already outnumber the humans on many instances, and AI agents are arriving on top of them. All of them hold real responsibilities. KeyForge governs them under **the same rules, the same severities and the same exception process** as employees.

AGENT SEGREGATION OF DUTIES An agent's authority is evaluated, not assumed Every agent is registered with its vendor, its model and the tools it is entitled to call. Delegation is explicit — which human delegated authority, for what, and until when — and an agent whose delegator leaves the organisation loses its authority with them. Read-only agents are separated from those that can write or execute, because only the second group carries duty-segregation risk.	MCP SEGREGATION OF DUTIES A tool server is an entitlement bundle Grant an agent supplier creation on one MCP server and payment release on another, and you have built a toxic combination that exists in neither system. EBS sees one integration account behaving normally; the conflict lives in the composition of tools above it. KeyForge evaluates the agent's effective capability — every tool it can call, the identity it acts as, and everything it inherits by delegation — against the same rule library used for people.
RISK AT THE MCP LAYER Enforced at call time, not discovered afterwards The gateway checks the request as the agent makes it. A call that would complete a toxic combination is denied and recorded, with the rule that blocked it. That turns the agent layer from an unmonitored route around your EBS controls into an enforcement point producing the same evidence as every other one.	JUST-IN-TIME CRITICAL ACCESS Elevated access for a purpose and a window Critical EBS access is requested for a stated reason and a bounded period rather than held permanently — for people, for service accounts and for agents alike. Just-in-time eligibility and vault brokerage are tracked per account, sessions are logged and assessed, and use is reconciled against the reason given. Standing privilege becomes the exception that has to be justified.

Same rules. Same evidence. Same audit record — whether the requester is an employee, an integration or an agent.

Five steps from a read-only extract to a scored violation register

An ordered sequence in which each step consumes the previous one's output. A first EBS instance typically produces scored findings inside the opening week.

01	Read-only extract Catalogued queries collect users, responsibilities, and who holds what — optionally reaching down to individual functions and menus. Nothing is written back to EBS.
02	Resolve the operating units The security profile behind each responsibility is followed through to the operating units it actually grants, so every assignment arrives carrying its real scope. Responsibilities with unrestricted org access are recorded as global.
03	Reconcile against the current picture The batch is merged into the live model, responsibilities that have disappeared from the source are retired, and every record is stamped for traceability back to the extract it came from.
04	Generate business functions Each responsibility becomes a business function automatically. Finer-grained functions are defined by hand only where a rule genuinely needs to target one specific screen or action.
05	Author rules, then analyse Same-operating-unit rules for conflicts inside EBS, scope-independent rules for cross-application ones, then a scored run. Results land in the violation queue, the role-risk view and the simulation console together.

EBS onboards the same way as every other platform

Only the extract mapping and the data dimension differ. Start with EBS and add platforms as the programme grows — the rules, the controls, the exceptions and the audit record all carry across, because a rule never needs to know which system each side came from.

PLATFORM	ACCESS IS GRANTED BY	RESOLVED TO	DATA DIMENSION
Oracle EBS	Responsibility	Menu and form function	Operating unit
Oracle Fusion	Job and abstract role	Duty role, then privilege	Business unit, ledger
SAP	Single and composite role	Object, field and value	Company code, plant
Workday	Security group	Domain and business-process policy	Organisation
PeopleSoft, Dynamics 365, NetSuite	Role	Permission list, duty, privilege	Business unit, legal entity
SaaS and custom applications	Application role or profile	Permission, via configurable extract	As modelled

PLATFORM	ACCESS IS GRANTED BY	RESOLVED TO	DATA DIMENSION
Machine identities	API permission, agent tool grant	Callable operation	Environment

FITS YOUR LANDSCAPE

Runs alongside the identity platform you already own

Read-only from EBS	The instance is read through catalogued extracts and nothing is written back, so a risk assessment can start without a provisioning integration, a change window, or anything installed in EBS.
A decision your request workflow can call	The request-time conflict check answers in-line, so any access request system — your IGA suite, your service desk, or a home-grown workflow — can gate an approval on it without adopting a second rule matrix.
Findings pushed where your team already works	New high and critical findings are emitted as events, so they land in your monitoring, ticketing or alerting systems as they arise instead of waiting to be collected from another console.
Queryable by your own agents	The risk engine is exposed as a governed tool surface, so an approved assistant can pull a dashboard, trace a violation, run a what-if simulation or file an exception request — under the same permissions and audit trail as a person.