

RED BEAR ARTIFICIAL INTELLIGENCE

ABU DHABI, UNITED ARAB EMIRATES

Enterprise Ontology for Inter-Entity Automation

Why automation across the boundary fails without a shared ontology, and how to build one that holds

Automation between entities breaks at the seams: the hand-off nobody owns, the wait nobody modelled, the data one side reads and the other side changed. This paper shows how the coordination structures of an enterprise ontology make those seams explicit, and how Red Bear EOM finds, checks and traces them.

Empower. Trust. Verify.

White paper EOM-WP-02 · Red Bear white paper series · Version 0.1

26 September 2026

Summary

Automation inside one entity is an engineering problem. Automation between entities is an ontology problem first. When a licence cannot issue until a fee is confirmed by a shared payment service, when a case in one department waits on an inspection by another, when a decision reads a register that a third entity maintains, the dependency is not an interface: it is a wait, an initiator or an access relationship between transaction kinds that belong to different organisations. Unless those relationships are modelled, agreed and kept current, the automation built on them will be brittle, and nobody will be able to say why. This paper explains the three coordination structures of the DEMO methodology, how they cross entity boundaries, and how Red Bear EOM discovers them from documents and connectors, checks them with people, and traces what happens when one of them changes.

WHO SHOULD READ THIS

- Heads of digital services and shared platforms
- Integration and automation architects
- Entity architects whose processes depend on other entities
- Process owners in shared-service and government-ERP programmes

WHAT YOU WILL BE ABLE TO DO AFTERWARDS

- Name the three kinds of dependency between functions and recognise each in your own processes
- Explain why a wait link is not an interface, and what that means for automation design
- Read a function card and a dependency trace
- Specify what a connector may and may not do in your tenant

1 What it is

Every result an organisation produces is created by a **transaction**: one actor role requests it and one actor role executes it, following a universal pattern of request, promise, declare and accept, with decline, reject and revoke as the exceptions (Dietz and Mulder, Ch 4 §4.4.4). A transaction kind with its executor role is a **transactor role**, and it is what a business user means by a function. The DEMO methodology connects transactor roles by exactly three **coordination structures** (Dietz and Mulder, Ch 4 §4.4.6; Ch 9 §9.3.2):

Structure	Link	Plain reading	Influence mode
Interaction	Initiator link	Who may ask for this result. Initiator links always form trees, one tree per kind of case	Activating
Interimpediment	Wait link	This function cannot proceed until a transaction of another kind reaches a given state	Impeding
Interstriction	Access link	This function may read the facts that another transaction kind has created; what it can see limits what it may decide	Restricting

The three modes on the right come from PRISMA, the theory of discrete-event systems that underlies DEMO: a processor can activate, restrict or impede another (Dietz and Mulder, Ch 4 §4.4.5). Red Bear EOM pairs each link kind with its mode, and that pairing is what turns a static diagram into something that can be traced.

Inside one entity these links are ordinary process design. Across entities they are the seams of the federation. Interaction trees are autonomous processes, each with its own case kind, and DEMO is explicit that separate trees are coordinated by wait links rather than by one seamless flow (Dietz and Mulder, Ch 4 §4.4.6). A licensing process in one department and a fee collection process in a shared finance service are two trees; the licence's dependence on the fee is a wait link between them. That wait is the real integration requirement. The API, the message queue and the batch file are implementations of it, and they will be replaced; the wait will not.

DEMO adds one more distinction that matters for automation: the **layers**. Only *original* acts, those that decide, judge, create or observe, change the production world and need a human with authority. *Informational* acts (remembering, recalling, computing), *documental* acts (saving, providing, transforming documents) and *material* acts (storing, transmitting, copying) support the original ones and may be delegated to technology, with humans still accountable (Dietz and Mulder, Ch 4 §4.4.7). Inter-entity automation is almost always automation of the I, D and M layers in service of O-layer functions that stay human. Knowing which is which tells you what may be automated and what must not be.

Key points

- A function has three kinds of dependency: who asks, what it waits for, what it reads
- Across entities, those dependencies are the integration requirement; interfaces are their implementation
- Only original acts change the world; informational and documental support can be automated

2 Why it matters

Governments and groups are building shared platforms at pace: one ERP for every department, one identity service, one payment gateway, one case management product licensed group-wide. Each promises automation across the boundary. Each depends on relationships that were never written down, because they lived in the heads of the people who ran the manual version.

The failure modes are predictable. A shared service changes its process and a downstream entity's function silently stops being requested. A register is migrated and every function that read it now decides on stale facts. A wait that used to be enforced by a person carrying a file becomes a race condition. Dietz and Mulder's Post Office case is the extreme form: the Horizon system's outputs were treated as original facts and the human transaction states that would have allowed revocation and contest were unreachable in practice (Dietz and Mulder, Ch 10). The childcare allowance scheme shows the same pattern in a government setting: decline, reject and revoke were unreachable, and parents became the de facto executors of evidence gathering (Dietz and Mulder, Ch 12 §12.3).

The economic case is as strong as the civic one. Transaction cost theory, which DEMO extends into an enterprise cost model, treats the cost of coordination as real cost (Dietz and Mulder, Ch 7 §7.1–7.2). Every undocumented wait between entities is coordination cost paid daily by people who reconcile, chase and re-key. An ontology makes the cost visible, and automation designed on the ontology removes it without removing the accountability.

Automation intent	What it actually depends on	What breaks when it is not modelled
Auto-create an entity in EOM when it is onboarded to the shared Gov ERP	A response link from the ERP's onboarding transaction to EOM's entity registration	Entities exist in the ERP that the group's architecture cannot see
Issue licences without manual fee checks	A wait link from *fee confirmed* to *licence issued*, with its state and cardinality	Licences issue before payment, or block forever when the fee service is down

Automation intent	What it actually depends on	What breaks when it is not modelled
Populate job descriptions from HCM and the directory	Access links from the position-building function to the HCM and GAL banks, with verification	Stale titles and unreachable contacts published as fact
Route a change for approval automatically	The layer of the function changed (O or not) and the entities it is realised in	Governance decisions taken as IT tickets

3 The challenges

Challenge	Why it is hard	What goes wrong without a method
Discovering the dependencies	Waits and reads are rarely documented; they are visible in process documents as a sentence, in Fusion as a configuration, in the directory as a job title, in Teams as a message	Integration is designed from the API catalogue, not from the dependencies
Capturing them with their state and cardinality	A wait is on a specific state (accepted, declared) with a cardinality (1..1, 0..*); getting this wrong changes the behaviour	Automation waits on the wrong state or on the wrong number of things
Maintaining them across two owners	Each end of a cross-entity link belongs to a different entity, with its own change cycle	One end changes; the other finds out in production
Understanding the impact of change	An impediment propagates: a delay upstream blocks execution downstream, then blocks the enclosing transaction, then delays whatever it initiates	Impact is assessed at the interface, not along the chain
Driving standardisation	Standardising a wait (everyone waits for fee confirmation the same way) is valuable; standardising an original act (everyone decides the same way) removes judgement (Dietz and Mulder, Ch 6 §6.3)	Automation removes the human where the human was the point
Knowing who decides	Changing a cross-entity link changes the essential model of two entities	Neither board approves it, or both block it

4 How Red Bear EOM helps

Challenge	What EOM does	Module
Discovering	Claude reads process documents, minutes and questionnaire answers and proposes initiator, wait and access links, each marked inferred with the passage it came from; connectors read the organisation from Oracle Fusion HCM, suppliers and spend from Procurement and Financials, projects from PPM, and people and context from the Microsoft 365 directory, SharePoint, Viva and, by opt-in, Teams	CN Connectors, KE Knowledge
Capturing	Each wait and response link carries its awaited state, its attachment point (order, execution or result phase) and its cardinality; each transactor role carries its layer; the DEMO rule of one product kind and one executor is enforced	LO Ontology (FR-LO-05)
Maintaining	Both ends of a cross-entity link are owned; a change to either raises a reassessment task for the other; source monitors watch the documents and configurations the link was found in	UN Understanding

Challenge	What EOM does	Module
Impact of change	The dependency trace walks the typed links with one effect rule per kind: blocked, cannot complete, requested late, decides on wrong information, implementation affected; confidence falls through inferred links; the trace is stored and re-runnable	VZ Visualisation (FR-VZ-09)
Standardisation	The layer rule: an effect in I, D or M is never reported as a change to what the organisation decides; patterns for shared services and the anti-pattern *automated decline without human escalation* are in the catalogue	RP Recommendations
Who decides	A trace that touches an O-layer element in more than one entity is marked governance and can be sent to impact analysis or the governance queue	GV Governance (FR-VZ-10)

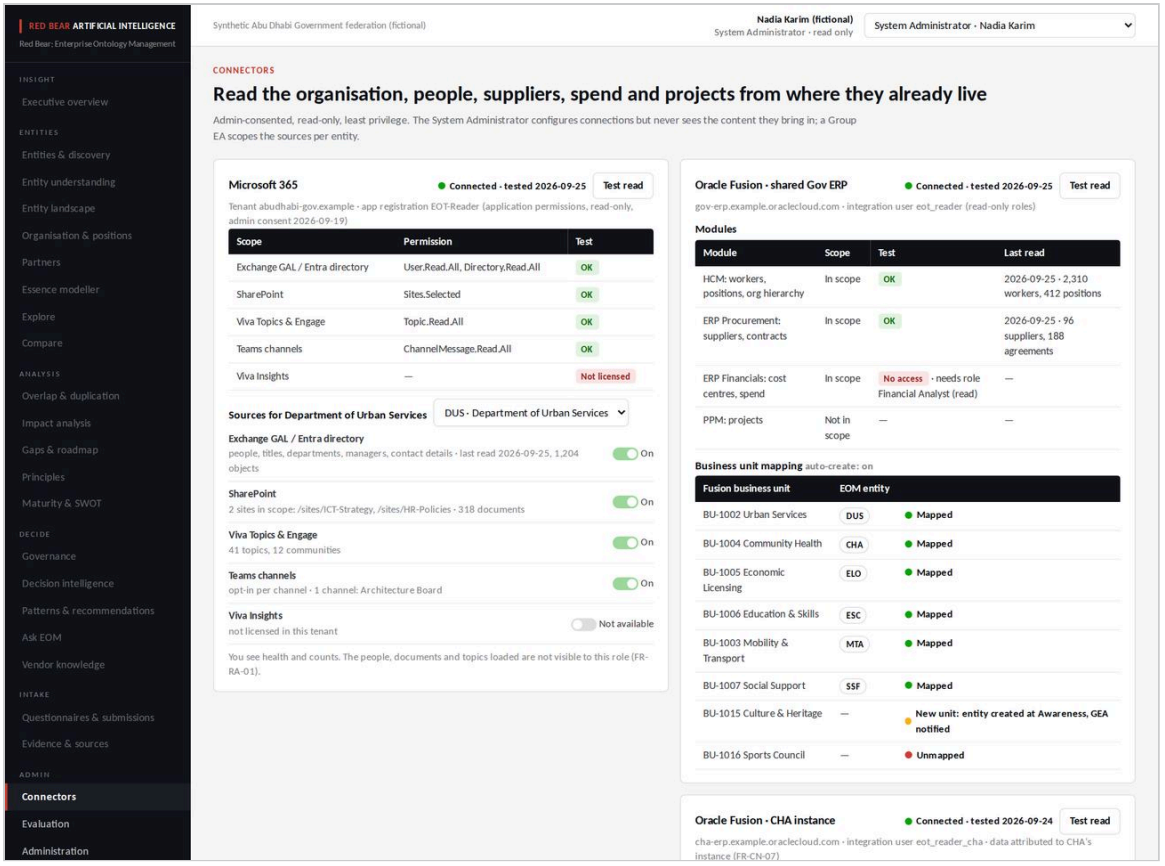
Design rule

Model the wait, not the interface. An interface is a way of implementing a wait, an initiation or a read. EOM records the dependency at the essential level and links each entity's implementation to it, so the automation can be replaced without losing the requirement.

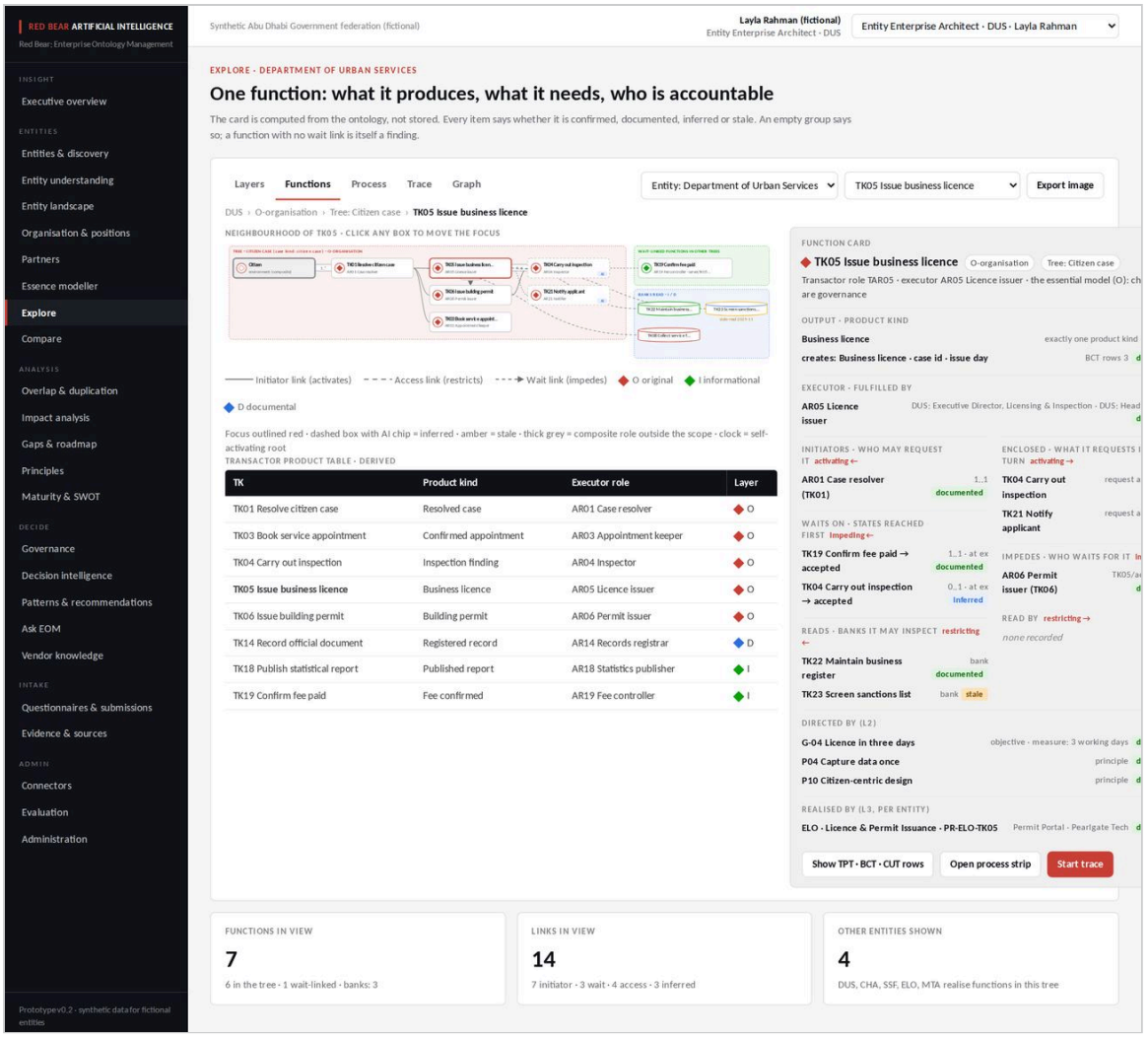
5 How it is used

Step	Who	What happens
1 Connect	SA, customer admin	Entra app registration with admin-consented, read-only application permissions for the Microsoft 365 sources; a read-only integration user on each Oracle Fusion instance, including the shared Gov ERP; the System Administrator configures, never reads
2 Discover the organisation	EEA	HCM business units and positions, GAL job titles, SharePoint sites and Viva communities are read and mapped to the entity; contacts are verified against a live account before they are shown as reachable
3 Find the functions and their links	CON, EEA	Claude proposes transactor roles and links from the documents and the connector data; the Entity EA confirms, corrects or rejects each; the Coordination Structure Diagram fills in
4 Agree the cross-entity links	GEA, both EEAs	Every link whose ends belong to different entities is listed; both Entity EAs confirm it; the Group EA records the awaited state and cardinality as the agreed integration requirement
5 Trace before you build	Integration architect	For each planned automation, run a trace from the function it touches under *changed*, *delayed* and *removed*; attach the trace to the design decision
6 Keep it current	All	Source monitors and connector re-reads raise reassessment tasks when a link's source changes; the trace is re-run and the two effect lists compared

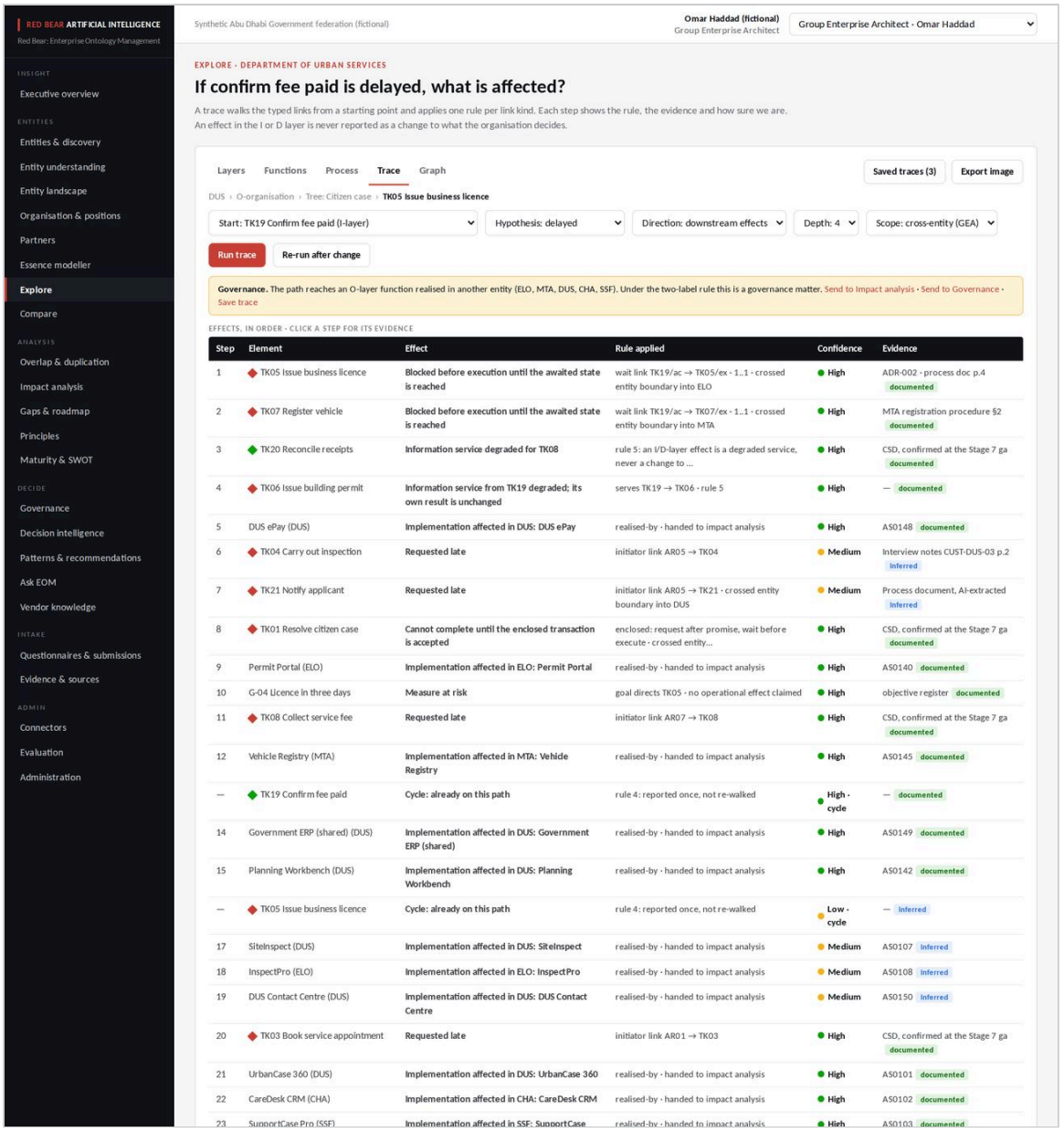
6 The user experience



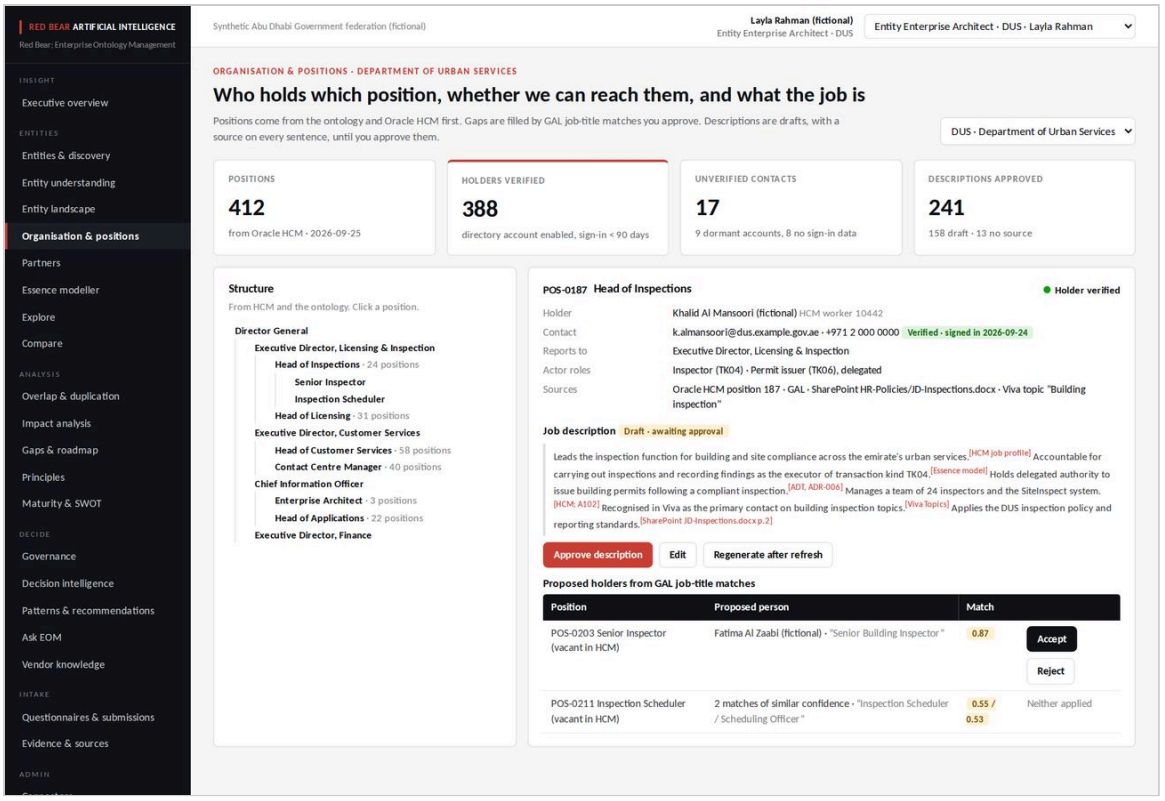
S-23 Connectors: S-23 Connectors. Read-only, admin-consented connections to Microsoft 365 and Oracle Fusion, including the shared Gov ERP and per-entity instances. The System Administrator sees connection state and scope, never content. Prototype render on synthetic data for fictional entities.



UC-40: S-19 Explore, Functions mode. The function card for one transactor role: its single product kind, its executor and the positions that fulfil it, and the three inbound groups (initiators, waits-on, reads) and three outbound groups (enclosed, impedes, read-by), each item marked documented, inferred or stale. An empty group is shown as "none recorded". Prototype render on synthetic data for fictional entities.

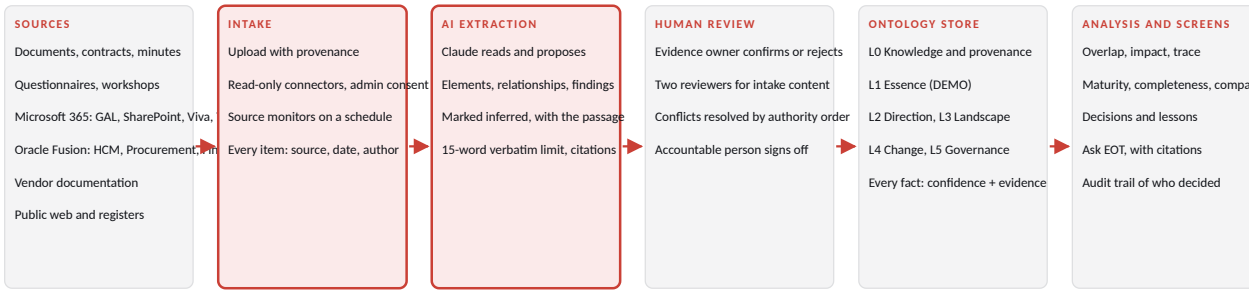


UC-41: S-19 Explore, Trace mode. "If fee confirmation is delayed, what is affected?" The ordered effect list with the rule and evidence per step, the highlighted path on the Coordination Structure Diagram, the governance flag because two entities are reached, and the Process Structure Diagram strip showing exactly where the wait attaches. Prototype render on synthetic data for fictional entities.



S-24 Organisation & positions: S-24 Organisation & positions. Positions from the ontology and Oracle HCM first, gaps filled by directory job-title matches the Entity EA approves, contacts shown only when verified against a live account, and job descriptions held as drafts with a source on every sentence. Prototype render on synthetic data for fictional entities.

7 Where the data comes from, and how it is checked



The pipeline, with the intake and AI-extraction stages highlighted. Connectors and documents feed extraction; extraction proposes links; people confirm them.

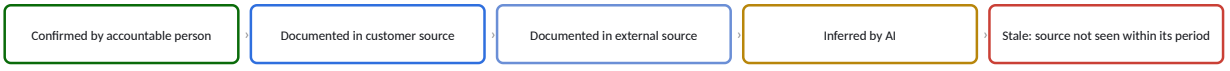
What each connector reads, and what it may never do

Source	Read	Used for	Never
Exchange GAL	Users, job titles, departments, managers, account state, last sign-in	Positions and contacts; verification (enabled account, sign-in within 90 days)	Mailbox content; write-back
SharePoint	Site and document metadata, documents in configured sites	Process documents, standards, minutes for extraction	Sites outside the configured scope

Source	Read	Used for	Never
Viva	Communities, topics, published profiles	Context and expertise for positions	Private conversations
Teams (opt-in)	Channel messages in channels an owner has opted in	Evidence for decisions and changes	Chats; channels without opt-in
Oracle Fusion HCM	Business units, positions, assignments	Organisation and positions per entity	Pay, personal data beyond name and position
Fusion Procurement and Financials	Suppliers, contracts, spend by category	Vendors and duplicated spend	Payment execution
Fusion PPM	Projects, budgets, milestones	Initiatives and roadmap	Approvals
Shared Gov ERP	Entity onboarding events	Auto-create Managed entities as candidates for review	Silent creation without review

How a link is checked

- **Proposed with its passage.** Every link Claude proposes carries the sentence or configuration it was read from, limited to fifteen words, with the source, page and date.
- **Confirmed by the owner of each end.** A link inside one entity is confirmed by that entity's EA; a link across entities by both, and the Group EA records the agreed state and cardinality.
- **Validated against the DEMO rules.** One product kind and one executor per transaction kind; a wait or response link must attach to a valid state; a leaf function that yields more than one product kind is flagged for decomposition.
- **Verified contacts only.** A person is shown as reachable only if their account is enabled and has signed in within the verification window; otherwise the position is shown with *contact unverified*.
- **Confidence follows the weakest link.** In a trace, every effect reached through an inferred link is one level less certain than the effect before it, and the plain-language explanation says so.



The five confidence levels, carried by every link and shown on every diagram.

8 Benefits realised

Benefit	How it shows up	Measured by
Integration requirements that survive re-platforming	The wait, initiation and read are recorded at the essential level; the interface is a realisation	Cross-entity links with both ends confirmed
Fewer production surprises	Every planned automation has a trace attached under three hypotheses	Share of integration changes with a stored trace
Coordination cost removed, accountability kept	I, D and M support automated; O-layer functions stay with named people	Functions by layer; automation candidates by layer
Organisation and contacts that are real	Positions from HCM and the ontology; contacts verified before they are shown	Verified share of positions per entity

Benefit	How it shows up	Measured by
Shared-platform onboarding that the group can see	Gov ERP onboarding creates a candidate entity for review	Candidates from ERP events; time to review
Governance that catches cross-entity change	Traces reaching two entities are flagged and routed	Governance-flagged traces; decisions with a trace attached

9 Risks managed

Risk	Control in Red Bear EOM
Connector over-reach and privacy	Admin consent; application permissions limited to read; configured scopes per source; Teams by channel opt-in only; the System Administrator never sees content; no write-back to any source (D-E5)
Wrong state or cardinality on a wait	Both recorded explicitly and validated; the PSD strip shows the attachment point so a reviewer can see it; traces test the link under *delayed* and *unreachable*
Automating an original act	Every function carries its layer; the trace never reports an I/D/M effect as a change to the production world; the anti-pattern *automated decline without human escalation* is in the catalogue
Inferred links treated as agreed	Inferred links are dashed with an AI chip and cannot be cited; cross-entity links need confirmation from both ends before they are used as requirements
Stale contacts and titles published	Verification rule (A-E11); unverified contacts withheld; job descriptions held as drafts until approved
One entity's change breaking another	Both ends owned; reassessment task raised for the other end; re-runnable traces compare effect lists before and after
AI processing outside the UAE	Claude on the Anthropic API in development and in Microsoft Foundry inside the Azure tenancy for live; neither UAE-resident today, stated as NFR-12 and mitigated by data minimisation, no training on customer data, and the option to route only extraction prompts, never stores, to the model

Sources

Every definition and every claim attributed to a source in this paper names its source in the text, in the form *author, work, chapter or section*; the full references are below. Statements without a citation are Red Bear's own position, developed for Red Bear: Enterprise Ontology Management, and are marked RB-ORIGINAL in the product's method. Quotations are limited to fifteen words.

Cited as	Source	Authority
TOGAF Standard	The Open Group, *The TOGAF Standard*, 10th Edition, C220 Parts 0–5 (2022), used under Red Bear's licence	Industry Framework
Dietz and Mulder	J. Dietz and H. Mulder, *Enterprise Governance and Management*, Springer (2026)	Academic Reference
Greefhorst and Proper	D. Greefhorst and E. Proper, *Architecture Principles*, Springer (2011)	Academic Reference
Agile EA Playbook	The Open Group, *The Agile Enterprise Architect Playbook*, G226 (2022)	Industry Framework
Azure Architecture Center	Microsoft, *Azure Architecture Center* (19 patterns, read 25 September 2026)	Vendor Reference Architecture
Red Bear	RB-MM metamodel, RB-EAM method, EOM-FSD-01 v1.1 draft, function-explorer.md, eval-set.md	Red Bear

Glossary

Term	Meaning in this paper
Transactor role	A transaction kind together with its executor actor role; the unit the user sees as a function
Initiator link	Which actor roles may request transactions of a kind; forms trees (interaction structure)
Wait link	An actor must wait until a transaction of another kind reaches a given state (interimpediment structure)
Access link	An actor may inspect the facts of another transaction kind's bank (interstriction structure)
Response link	A state in one transaction triggers an act in another; DEMO-4 also calls it a causal link
Enclosed transaction	A transaction requested after the enclosing one is promised, and accepted before it executes
Influence mode	PRISMA's three ways one processor affects another: activating, restricting, impeding
Layer (O/I/D/M)	Original, informational, documental and material acts; only original acts change the production world
Function card	EOM's computed view of one function: output, executor, initiators, waits-on, reads, enclosed, impedes, read-by, directed-by, realised-by
Trace	A stored run of the dependency walk: start element, hypothesis, path, effects and confidence per step

The series

This paper is one of five. Each stands alone and each follows the same nine sections, so a reader who knows one knows where to look in the next.

Paper	Title	Theme
EOM-WP-01	Enterprise Ontology for Federated Organisations	One essence, many entities
EOM-WP-02	Enterprise Ontology for Inter-Entity Automation	Why automation across the boundary fails without a shared ontology (this paper)
EOM-WP-03	Enterprise Ontology Maturity and Baselining	How much do you actually know?
EOM-WP-04	Enterprise Architecture Roadmap Harmonisation	Roadmaps that agree
EOM-WP-05	Enterprise Architecture Decision Intelligence	Decisions with a memory

Red Bear Artificial Intelligence · Abu Dhabi, United Arab Emirates · info@redbearai.net · www.redbearai.net