

1 CMDB CORE FRAMEWORK

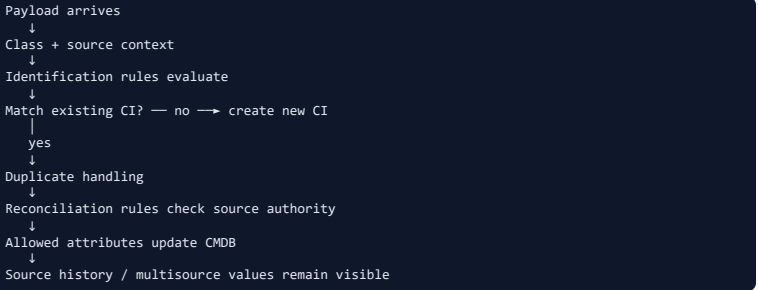
Term	Meaning
CMDB	Central store for infrastructure CIs and relationships.
CI	A row/record representing a tracked item.
Class / Table	CMDB table that stores similar CI types.
Attribute	Column/value describing a CI.
Relationship	Typed connection between CIs, users, or groups.
Parent / Child class	Hierarchy used for inheritance of attributes and some rule definitions.

- MUST KNOW**
- **Top of hierarchy:** Configuration Item [cmdb_ci].
 - **Independent CI:** can exist on its own.
 - **Dependent CI:** requires a qualifying relationship/context to exist.
 - **Principal class:** important class designation used by CMDB success advisor and CI filters.

CI = row Attribute = column Class = table Relationship = typed edge

2 IRE FLOW AT A GLANCE

Purpose: preserve CMDB integrity during create/update by identifying the target CI and applying reconciliation controls.



Step	What it decides
Identification	Is this an existing CI or a new CI?
Duplicate logic	Are multiple candidates matching?
Reconciliation	Can this source update these attributes?

EXAM TRAP
Identification and reconciliation are linked inside IRE, but they are **not** the same function. Many questions hinge on that distinction.

6 DUPLICATE CIS & DE-DUPLICATION

Property / fact	Key detail
glide.identification_engine.skip_duplicates	Default true .
glide.identification_engine.skip_duplicates.threshold	Default 5 .
glide.duplicate_ci_remediator.max.cis	Default 1000 .

- If **skip_duplicates = true** and duplicates are below threshold, IRE updates the **oldest** CI, which becomes the **main** CI.
- Other duplicates are tagged via **duplicate_of**.
- If **skip_duplicates = false**, matching fails with an error and none of the duplicates are updated.
- **De-duplication tasks are still created** in either case.
- Bulk remediation uses **template libraries**; template must be **published** before it can run.

- IMPORTANT LIMITS**
- Asset-related merges are not supported in the same way as standard CI merge data.
 - Lookup/related-table de-dup tasks require configuration in the allowed-class table.

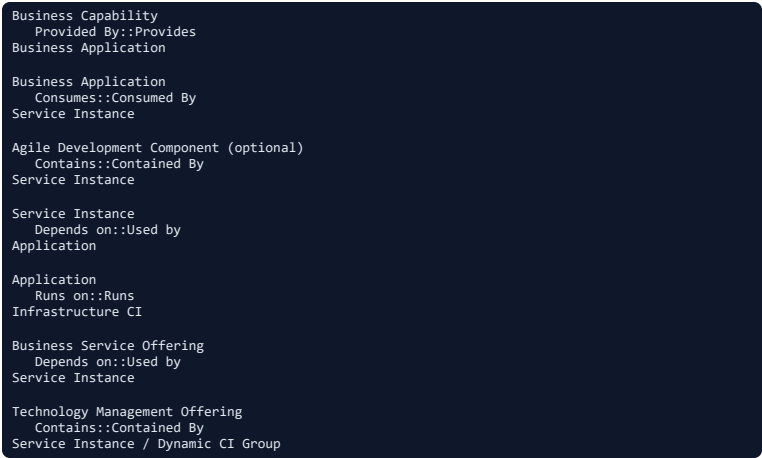
5 POPULATION & INTEGRATION PATHS

Method	What it does
Discovery	Discovers infrastructure CIs; commonly MID-based.
Service Mapping	Maps application/service topologies and dependencies.
Service Graph Connectors	Prebuilt third-party data imports into CMDB.
API SGC	Connector path for API-fed data.
IntegrationHub ETL	ETL-based integration/transformation path.
Integration Commons	Dashboard + RTE transforms + scripts to standardize values.
Manual Create CI (Workspace)	Creates CI while applying IRE processes.

- QUICK SETUP CHECKLIST**
- **Discovery:** MID server + credentials + schedules/patterns.
 - **Service Mapping:** map the service/application topology, not just raw devices.
 - **SGC Central:** manage connector lifecycle, monitoring, debugging.
 - **ACC:** agent-based collection can complement visibility strategy.

Specialty CIs such as some application service records may require dedicated wizards rather than the generic Create CI experience.

9 HIGH-VALUE CSDM RELATIONSHIPS



- | | |
|--|--|
| CONSUMER / SELLER VIEW <ul style="list-style-type: none">• Business Service• Business Service Offering | PROVIDER / TECHNOLOGY VIEW <ul style="list-style-type: none">• Technology Management Service• Technology Management Offering |
|--|--|

Business Application is not the operational deployed instance. **Service Instance** is the operational representation of the deployed system/app stack.

3 IDENTIFICATION RULES

- **Goal:** guarantee uniqueness of CIs during insert/update.
- **Inheritance:** child classes derive parent identification rules unless a more specific class rule exists.
- **Independent CI rules:** typically use attributes on the CI and, where configured, lookup/related data.
- **Dependent CI rules:** include the dependent relationship/hosting/containment context.
- **Create CI in Workspace:** classes without a valid identification rule are grayed out / unavailable.

High-value building blocks	Use
Identifier entries	Main matching criteria.
Lookup / related entries	Bring in related values for identification scenarios.
Class dependency metadata	Helps identify dependent CIs correctly.

MEMORY HOOK
Weak identifiers = duplicate flood. If many de-duplication tasks are created, suspect poor identification design first.

7 RELATIONSHIPS & GOVERNANCE

Purpose: keep relationship types and directions valid and consistent across CMDB tools.

Rule set	Use
Dependent relationship rules	Hosting / containment rules used by identification scenarios.
Suggested relationships	Guided relationship options used in the relationship editor.
Reference rules	Mostly used for valid referenced CI pairings.

- RELATIONSHIP TYPES YOU SHOULD INSTANTLY RECOGNIZE**
- Runs on::Runs Depends on::Used by Contains::Contained by Provided by::Provides
- Consumes::Consumed by Sends Data To::Receives Data From

If a relationship type/direction is blocked in a UI, check **relationship governance rules** before assuming the user lacks access.

11 DATA QUALITY, HEALTH & GOVERNANCE

CMDB HEALTH KPIS	
KPI	Examples
Completeness	Required/recommended fields missing.
Correctness	Orphan, stale, duplicate CIs.
Compliance	Certification / audit adherence.
Relationships	Orphan/duplicate relationships, hosting/containment/suggested relationship compliance.

- CMDB SUCCESS ADVISOR — DATA FOUNDATIONS**
- **Scope:** up to **50 principal classes**.
 - **Roles:** sn_cmdb_admin; pa_viewer for filtering; pa_data_collector for last-updated timestamp.
 - **Indicators:** missing name, location, managed by group, duplicates, not updated in 7 / 14 / 30 / 60 / 90 days, CI counts by principal class and by source.

- CMDB DATA MANAGER / CERTIFICATION**
- **Policy types:** Delete, Retire, Archive, Attestation, Delete CMDB Related Entry, Certification.
 - **Retire / Archive / Delete** require active retirement definitions for targeted classes.
 - **Data Certification** validates fields in CMDB and non-CMDB tables via scheduled/on-demand certification tasks.

REMEMBER THESE

- **Identification** answers “**what CI is this?**”; **Reconciliation** answers “**may this source update it?**”.
- **Child classes inherit** parent attributes/rules unless a more specific class-level rule/definition exists.
- **Business Application** is design/planning; **Service Instance** is operational.
- **Principal classes** drive Data Foundations scope and filter CI pickers in task records.
- **Create CI in Workspace** applies IRE; classes without a valid identification rule are unavailable.
- **Retire / Archive / Delete** policies require active retirement definitions for target classes.

4 RECONCILIATION RULES

Question answered: Which source is allowed to update which attributes?	
Concept	Meaning
Static reconciliation	Broad precedence/ownership for a class or class set.
Dynamic reconciliation	Conditional / filtered authorization for subsets of CIs or attributes.
Attribute governance	Prevents lower-authority sources from overwriting governed values.

- Use reconciliation to protect key values populated by trusted sources.
- Different sources may enrich different fields on the same CI.
- Reconciliation runs **after** identification finds the target CI.

- FAST CONTRAST**
- **Identification:** “Which CI?”
 - **Reconciliation:** “Which source wins?”

8 CSDM DOMAIN MAP

Domain	Key records	Operational?
Foundation	Business process, product models, contracts, locations, org data	No
Ideation & Strategy	Planning Item, Product Idea	No
Design & Planning	Business Capability, Business Application, Information Object	No
Build & Integration	Agile Development Component	No
Service Delivery	Service Instance, Tech Mgmt Service/Offering, Application, Infra	Yes
Service Consumption	Business Service, Business Service Offering, Catalog	Yes
Manage Portfolio	Cross-domain overlay for ownership/portfolio view	Overlay

Foundation tables are referential data. They support the other domains but are not the same thing as relationship-heavy operational CMDB service models.

12 TROUBLESHOOTING + EXAM TRAPS

Issue	Likely cause	Immediate fix
Class unavailable in Create CI	No valid identification rule	Define/fix the class identification rule
Lots of duplicate CIs	Weak identifiers / poor source keys	Tighten identification rules; review source payload quality
Archive/Delete policy fails	No retirement definition or CI not in retired state	Create/activate retirement definition; verify state
Missing key CI attributes	Low integration coverage	Review Discovery patterns / connector coverage / ETL mappings
Relationship option blocked	Governance rules disallow type/direction	Check relationship governance metadata
Intelligent Search not working	Wrong instance language or property disabled	Use English instance; verify NLQ property

- ADDITIONAL INFO**
- **Business Service Offering** lives in **service_offering** and **depends on a Service Instance**.
 - **Technology Management Service** = provider/technology view; **Business Service** = consumer/seller view.
 - **Foundation domain** is critical referential data, but it is **not** the same as operational service topology.
 - **CMDB Health** focuses KPI-based health; **Success Advisor** focuses guided outcome-based improvement.
 - **Create CI in Workspace** applies IRE; **Query Builder starting node** is the first node dragged.