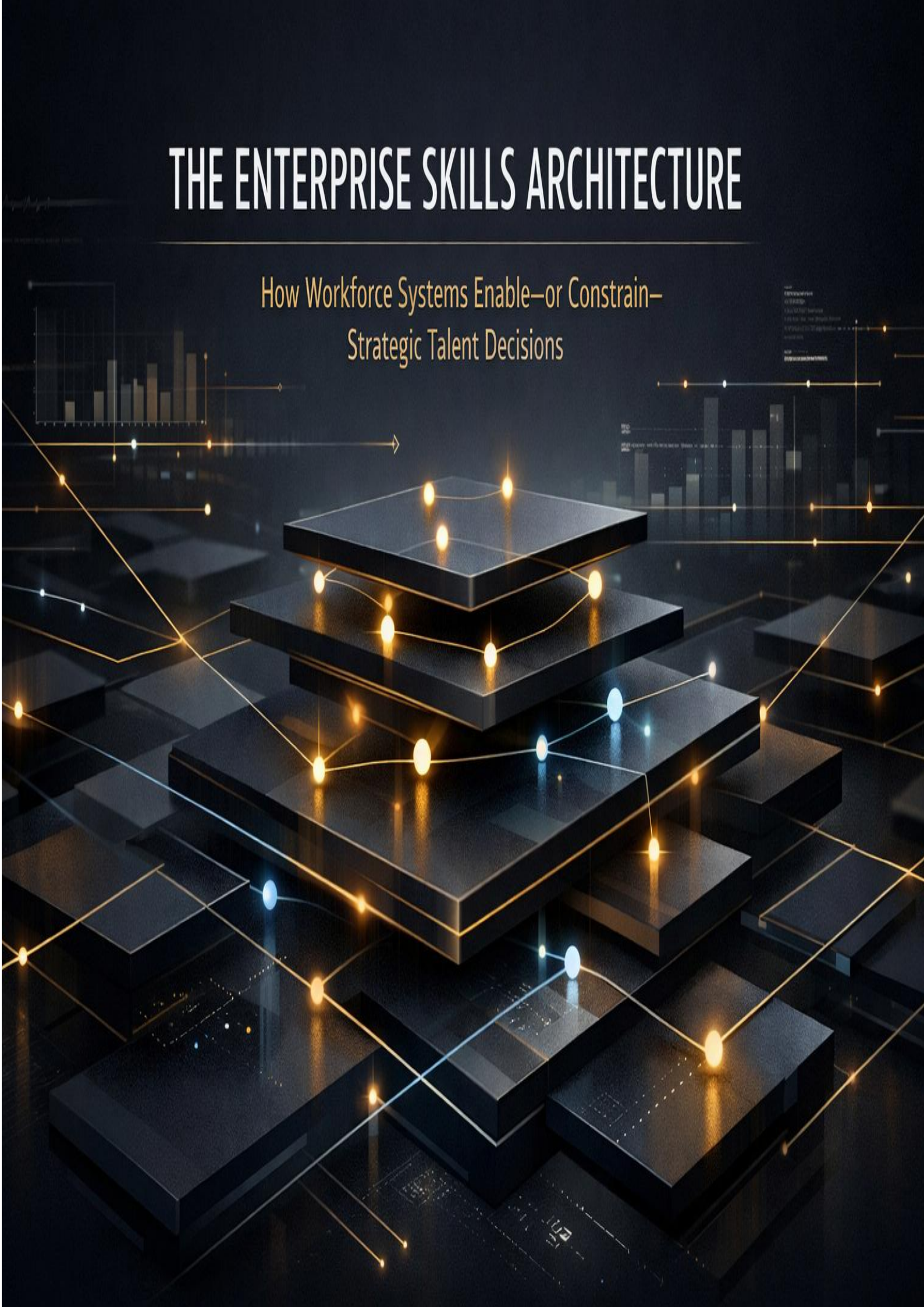
# THE ENTERPRISE SKILLS ARCHITECTURE

## How Workforce Systems Enable—or Constrain—Strategic Talent Decisions



# INDEX

# INTRODUCTION

Across large enterprises, the shift toward skills-based talent strategies is no longer aspirational—it is operational. Boards are asking for greater workforce agility. CEOs are demanding faster redeployment of talent. CHROs are under pressure to move beyond static job architectures toward a more dynamic, skills-driven view of capability. Yet beneath these ambitions lies a harder, less visible challenge: **the systems of record that power talent decisions were not designed for this transition.**

Human Capital Management (HCM) platforms—Oracle HCM Cloud, Workday, and SAP SuccessFactors—have evolved significantly over the past decade. They now support richer talent profiles, AI-driven inference, and broader integration ecosystems. However, **their foundational architectures remain largely job-centric**, shaped by historical requirements for compliance, reporting, and administrative control. As enterprises attempt to layer skills-based strategies on top of these systems, they often encounter fragmentation, semantic inconsistency, and limited interoperability.

This paper examines that reality head-on.

**Rather than approaching skills transformation as a conceptual or philosophical shift, we analyze it as a data and systems problem**. We take a deep, technical look at how leading HCM platforms define jobs, roles, skills, and competencies; how those objects relate to one another; how they integrate with external systems; and how each platform addresses the persistent challenge of skill taxonomy conflict and normalization.

The intent is not to declare one platform “better” than another. Each reflects a different architectural philosophy and set of trade-offs. The intent is to give enterprise leaders—particularly CHROs, SWP and TA leaders, and those accountable for workforce transformation—a clear, grounded understanding of **what these systems can and cannot do on their own**.

Ultimately, the shift to a skills-based organization does not happen inside the HCM alone. It requires an ecosystem approach—one that connects the HCM as the system of record with external intelligence and internal execution data. Understanding the underlying architecture is the first step toward building that ecosystem deliberately, rather than by accumulation.

The sections that follow break down this landscape in detail and lay the foundation for a more unified, data-driven approach to enterprise skills.

## 1. COMPONENT 1: JOB-RELATED DATA OBJECT MODELS

Most enterprise HCM platforms are fundamentally designed around the concept of a "job" as the core organizational construct. This job-centric model dictates how data is structured, related, and managed. The following sections provide a detailed, definitive data object model for Oracle HCM, Workday, and SAP SuccessFactors, focusing on the objects, relationships, and attributes that define a job.

## 1.1. ORACLE HCM CLOUD: THE PROFILE MANAGEMENT MODEL

In Oracle Cloud HCM, the Job Family and Job Profile (formally known as a **Model Profile**) are foundational components of the Workforce Structure. Together, they allow you to group similar roles and define the specific requirements (competencies, degrees, etc.) for those roles.

### 1.1.1. The Core Objects

In the Oracle data model, these are represented by specific business objects:

- **Job Family:** A high-level grouping of jobs based on similar functions or duties (e.g., "Engineering," "Finance," or "Human Resources").
- **Job Subfamily (Newer Feature):** A more granular categorization within a family (e.g., "Software Development" inside the "Engineering" family).
- **Job:** The generic role within the enterprise (e.g., "Senior Software Engineer").
- **Model Profile (Job Profile):** A collection of talent-related requirements associated with a Job. It defines *what* is needed to perform the job successfully.

### 1.1.2. Relationships and Hierarchy

The relationship is hierarchical and many-to-one in most directions, though a Job can only belong to one primary Job Family.

| Parent Object | Child Object | Relationship Type |
|---|---|---|
| Job Family | **Job Subfamily** | One-to-Many |
| Job Family | **Job** | One-to-Many |
| Job | **Model Profile** | One-to-One (usually) |
| Model Profile | **Content Sections** | One-to-Many (Competencies, Skills, etc.) |

- **Job to Family:** Each Job record contains a JobFamilyId which links it to its parent family.
- **Job to Profile:** The Job record is linked to a Model Profile (of type 'JOB'). While the Job defines administrative data (like Grade), the Profile defines talent data (like Skills).

### 1.1.3. Key Attributes

Each object carries specific metadata used for reporting and HR transactions.

**Job Family Attributes**

- Name & Code: Human-readable name and a unique system code.
- Status: Active/Inactive to control visibility in lists.
- Job Family Group: (Optional) An even higher level of grouping above the family.

**Job Attributes**

- Job Code & Name: The unique identifier for the role.
- Effective Start Date: Crucial for Oracle's "Date Effective" data model.
- Benchmark Job: A flag (Yes/No) indicating if this job is used for salary surveys.
- Valid Grades: A list of grades (e.g., G1, G2) that are allowed for workers in this job.

**Model Profile (Job Profile) Attributes**

Profiles are made of **Content Sections**, which are the most detailed part of the model:

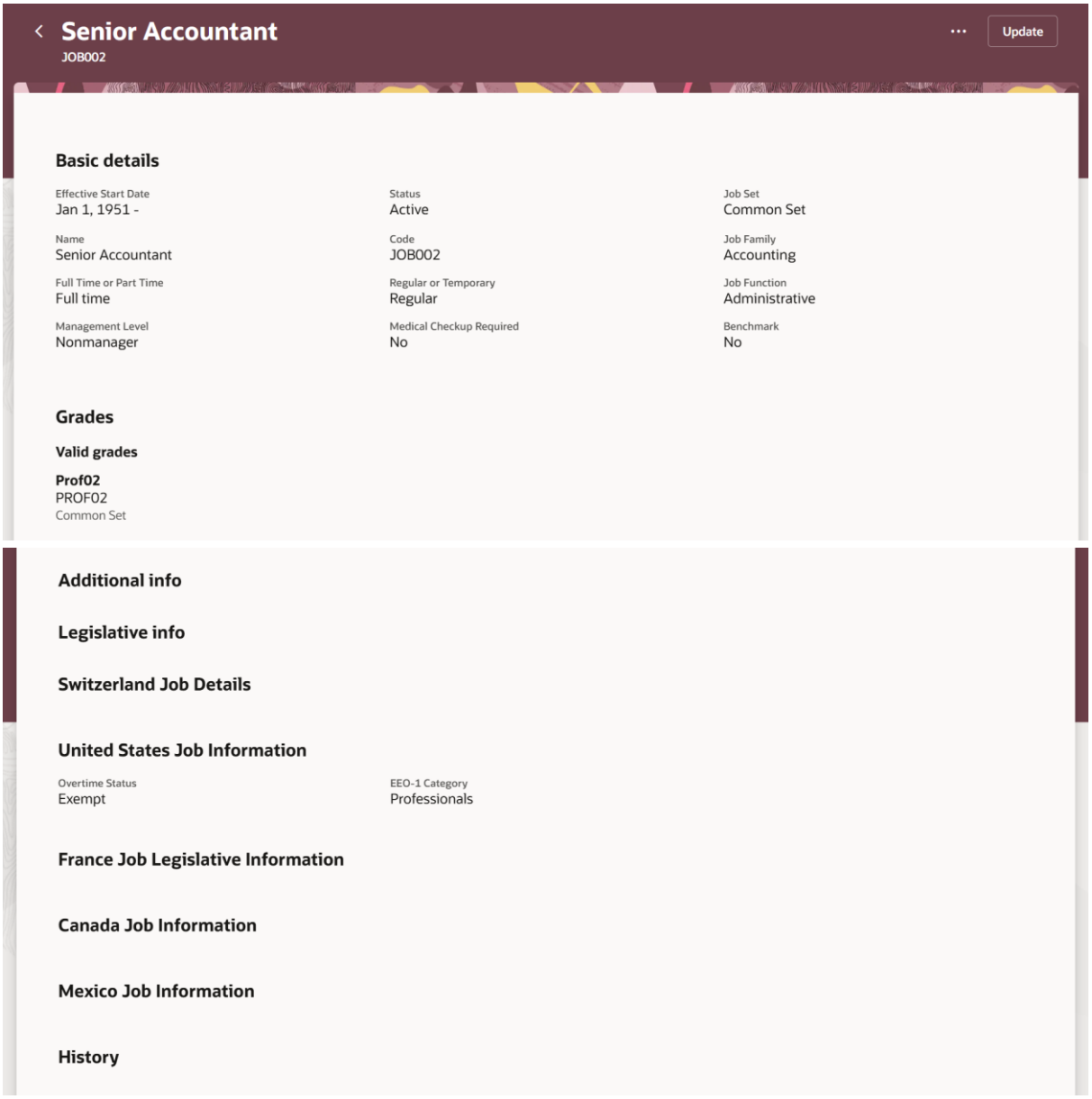
- Competencies: Specific skills required (e.g., "Critical Thinking") often with a "Required Rating."
- Degrees/Education: The minimum education level (e.g., "Bachelor of Science").
- Certifications: Specific licenses or professional certifications.
- Responsibilities: Text-based descriptions of the duties.
- Criticality: How vital the job is to the business (used in Succession Planning).

### 1.1.4. Summary of Data Flow

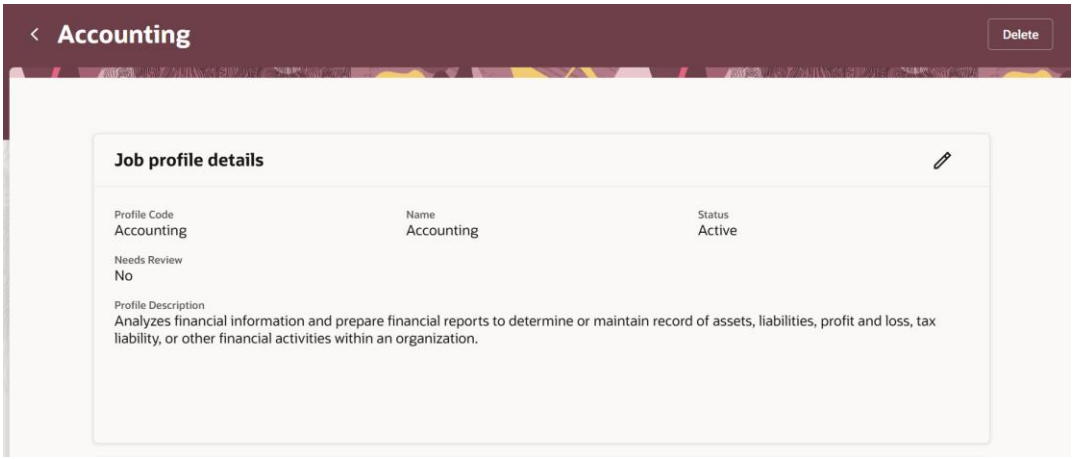
When a recruiter creates a job opening, they select the **Job**. Because the Job is linked to a **Job Family**, the reporting is automatically organized. Because the Job is linked to a **Model Profile**, the system automatically pulls in the "Must-have" competencies and qualifications for the job posting.

## 1.1.5. Screenshots:

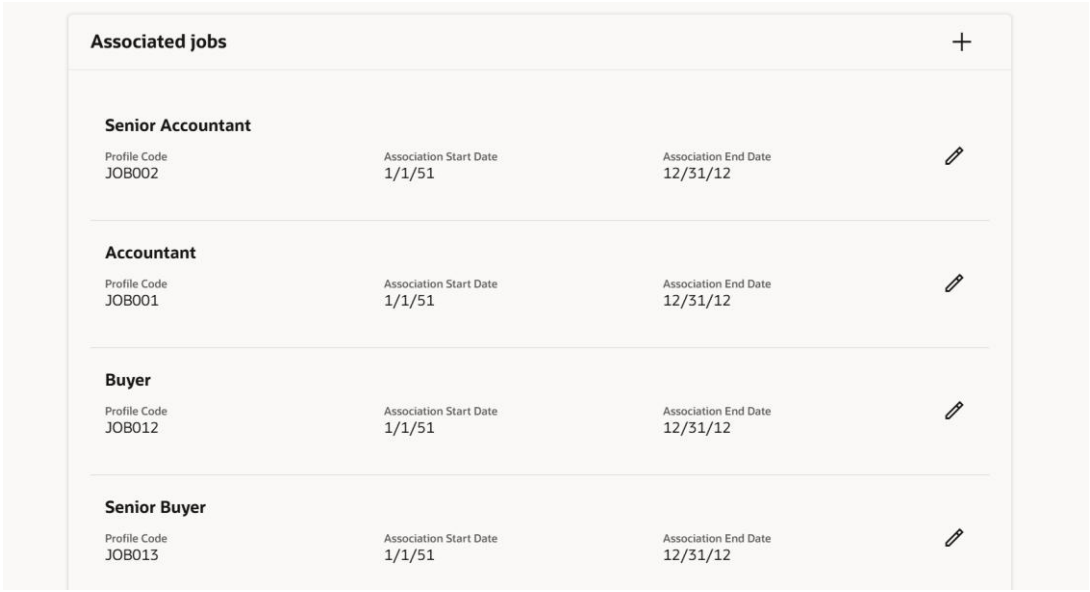
**Job**



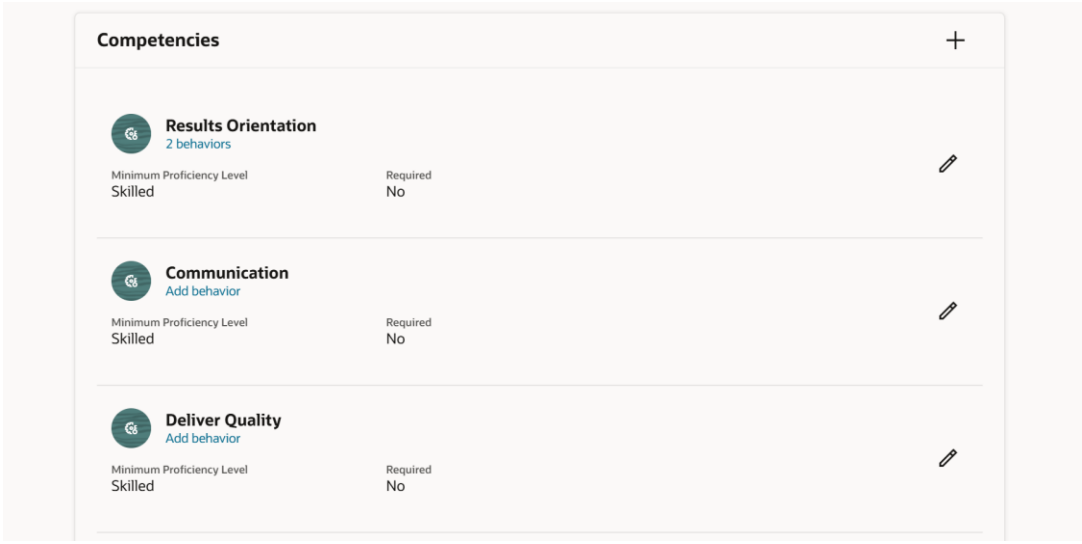
## Job Profile



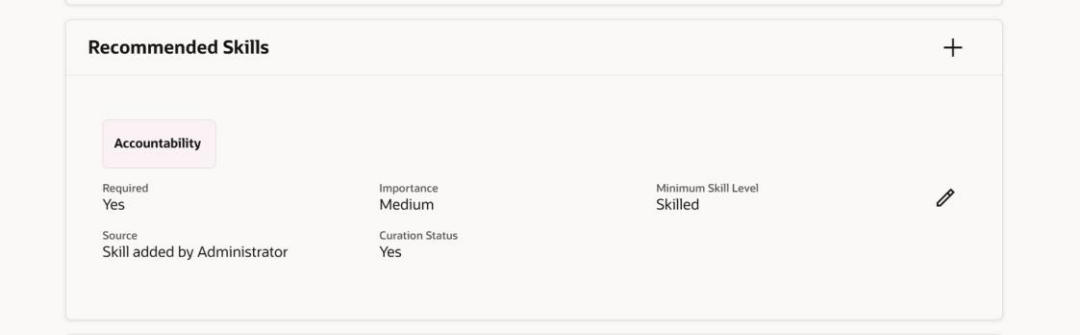

## Associated Jobs to the Job Profile




## Competencies attached to the Job Profile



**Skills attached to the Job Profile**




## 1.2. WORKDAY

Workday's architecture represents a significant departure from the traditional relational model. It is an in-memory, object-oriented system where data is accessed primarily through web services (SOAP and REST) rather than direct SQL queries. The core concept is the **Business Object**, and for jobs, the central object is the **Job Profile**.

### 1.2.1. Core Job Object: Job Profile

A Job Profile is a comprehensive object that encapsulates all the characteristics of a job, including its duties, responsibilities, and required competencies.

**Key Characteristics:**

- Object-Oriented: Data is treated as a collection of interconnected business objects, not as rows in tables.
- In-Memory**:** The primary data store is in-memory, which provides high performance for transactional operations.
- API-Centric: All interactions with the data model are through versioned web service APIs.

### 1.2.2. Core Job Architecture Components:

| Component | Description |
|---|---|
| Job Family Group | The highest level of job categorization (e.g., "Corporate Functions"). |
| Job Family | A collection of related jobs (e.g., "Finance," "Engineering"). |
| Job Profile | The foundational object, detailing a specific role's duties, responsibilities, and required skills. |

| | |
|---|---|
| Management Level | Defines the hierarchical level within a job family, supporting career progression. |
| Position | A specific instance of a job profile within a supervisory organization. |

## 1.2.3. Object Relationships:

The relationships in Workday are defined by the connections between its business objects, forming a hierarchical and networked structure:

**Worker → Position → Job Profile → Job Family → Job Family Group**

This chain illustrates the primary structural hierarchy. A worker occupies a position, which is defined by a job profile. That job profile belongs to a job family, which in turn is part of a job family group. Other critical relationships include:

- **Job Profile → Competencies/Skills**: This is the direct link from the job definition to the skills required to perform it.
- **Job Profile → Compensation Grade**: Connects the job to the organization's pay structure.
- **Position → Supervisory Organization**: Places the specific job instance within the managerial hierarchy.

This object-oriented model allows for a more holistic and less fragmented view of a job compared to a purely relational structure, with skills being an integral part of the Job Profile object itself.

## 1.2.4. Screenshots:

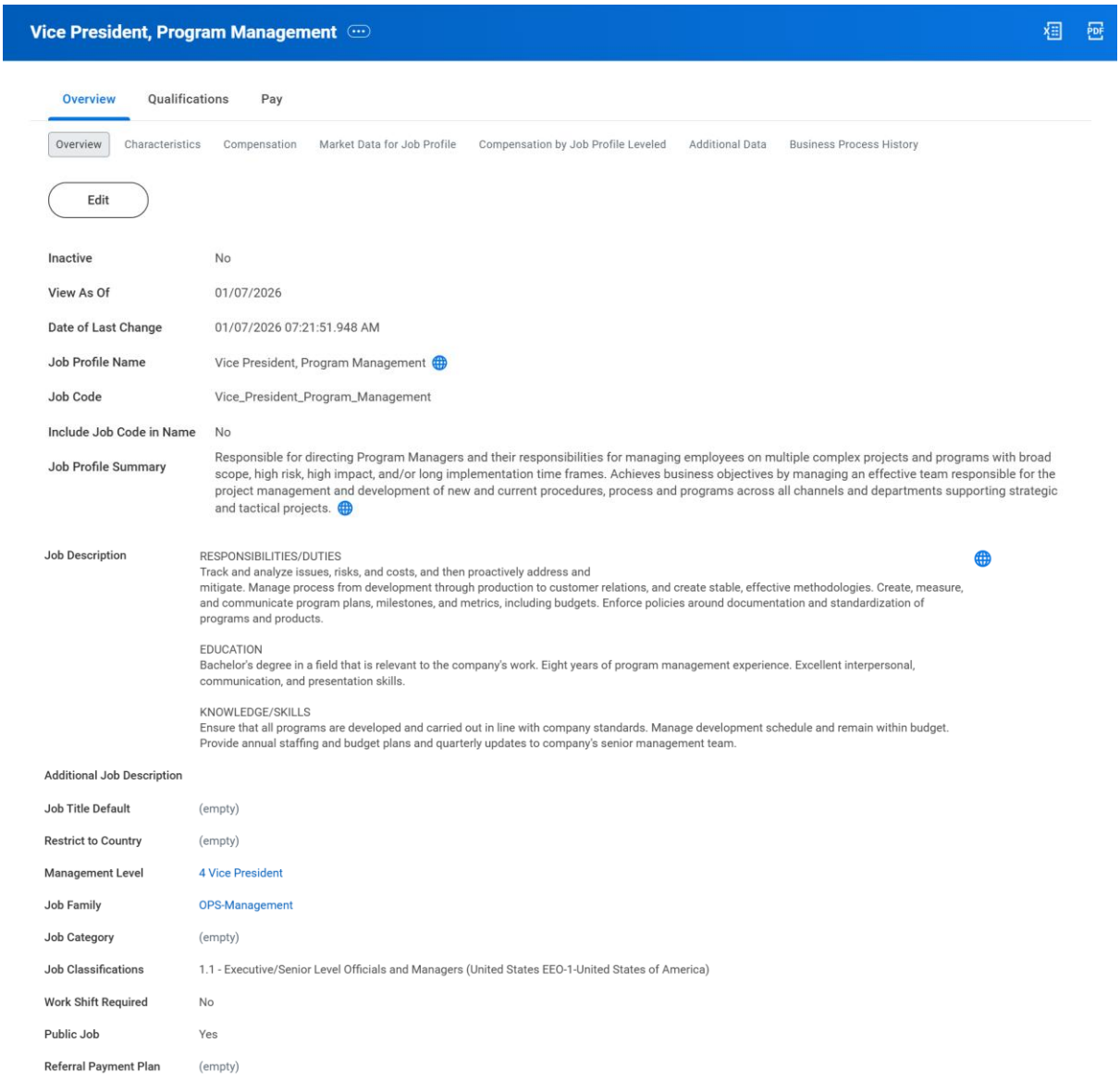
**Job Family Group**

## Job Family

**View Job Family** OPS-Management

| | |
|---|---|
| Job Family Name | OPS-Management |
| Job Family Summary | Operations Management |
| Part of Job Family Group | Operations |
| Job Profile | Vice President, Global Support<br>Vice President, Program Management<br>Vice President, Real Estate & Facilities<br>Vice President, Research & Development<br>Vice President, Support Operations |
| Inactive | No |

## Job Profile



**Vice President, Program Management**

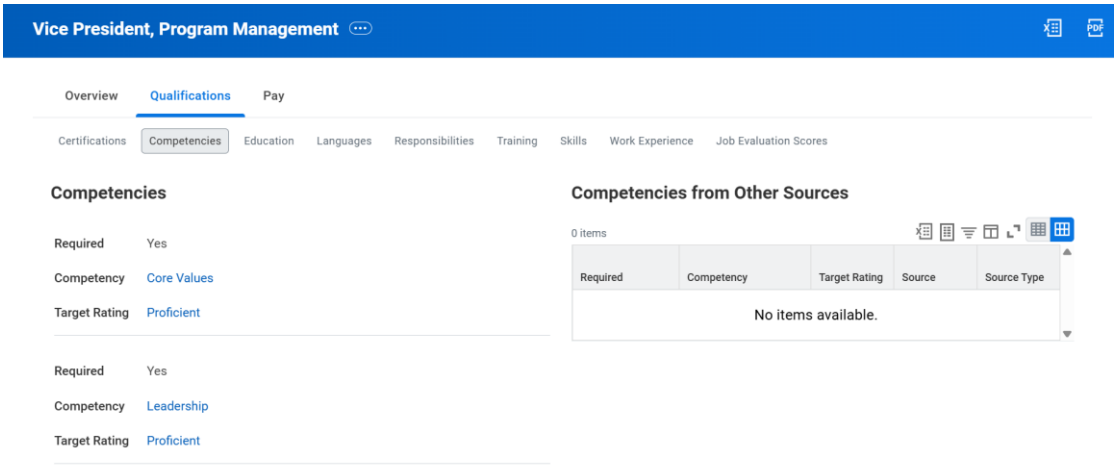
Overview | Qualifications | Pay

Overview | Characteristics | Compensation | Market Data for Job Profile | Compensation by Job Profile Leveled | Additional Data | Business Process History

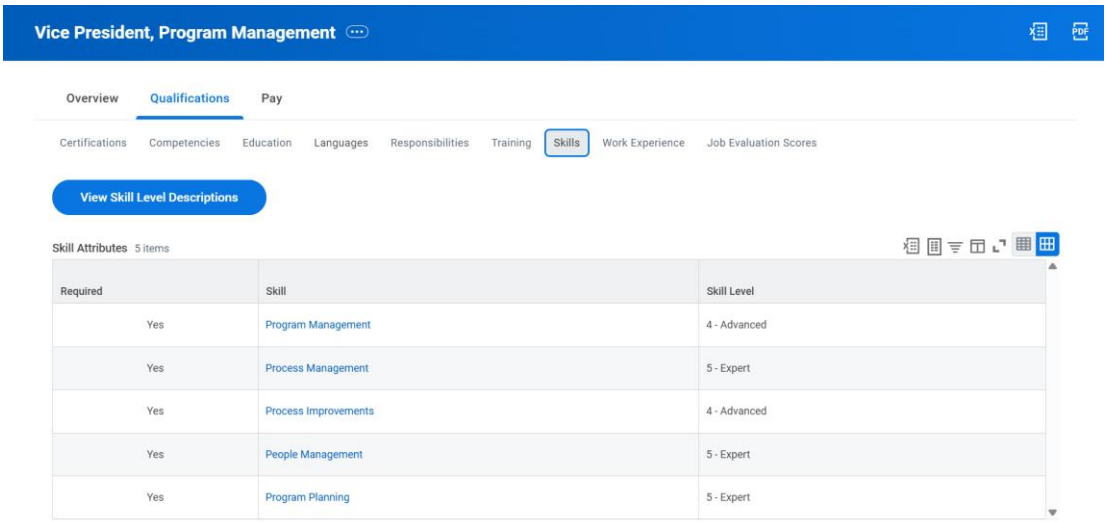
Edit

| | |
|---|---|
| Inactive | No |
| View As Of | 01/07/2026 |
| Date of Last Change | 01/07/2026 07:21:51.948 AM |
| Job Profile Name | Vice President, Program Management |
| Job Code | Vice_President_Program_Management |
| Include Job Code in Name | No |
| Job Profile Summary | Responsible for directing Program Managers and their responsibilities for managing employees on multiple complex projects and programs with broad scope, high risk, high impact, and/or long implementation time frames. Achieves business objectives by managing an effective team responsible for the project management and development of new and current procedures, process and programs across all channels and departments supporting strategic and tactical projects. |
| Job Description | RESPONSIBILITIES/DUTIES<br>Track and analyze issues, risks, and costs, and then proactively address and mitigate. Manage process from development through production to customer relations, and create stable, effective methodologies. Create, measure, and communicate program plans, milestones, and metrics, including budgets. Enforce policies around documentation and standardization of programs and products.<br><br>EDUCATION<br>Bachelor's degree in a field that is relevant to the company's work. Eight years of program management experience. Excellent interpersonal, communication, and presentation skills.<br><br>KNOWLEDGE/SKILLS<br>Ensure that all programs are developed and carried out in line with company standards. Manage development schedule and remain within budget. Provide annual staffing and budget plans and quarterly updates to company's senior management team. |
| Additional Job Description | |
| Job Title Default | (empty) |
| Restrict to Country | (empty) |
| Management Level | 4 Vice President |
| Job Family | OPS-Management |
| Job Category | (empty) |
| Job Classifications | 1.1 - Executive/Senior Level Officials and Managers (United States EEO-1-United States of America) |
| Work Shift Required | No |
| Public Job | Yes |
| Referral Payment Plan | (empty) |

**Competencies attached to the Job Profile**




**Skills attached to the Job Profile**




## 1.3. SAP SUCCESSFACTORS: THE DECOUPLED ROLE MODEL

SAP SuccessFactors utilizes a decoupled architecture where the definition of a role is managed independently from the physical position or the person performing it. This model is primarily driven by the Job Profile Builder (JPB), which serves as the "semantic core" by standardizing how capabilities and requirements are represented across the enterprise. Unlike legacy job-centric models, SuccessFactors abstracts the "what" of a job (attributes, skills, competencies) from the "where" (position hierarchy).

### 1.3.1. Core Objects and Relationships

The SuccessFactors model is built on the Metadata Framework (MDF), allowing for a flexible, object-oriented structure that can be localized and extended. The hierarchy typically flows as follows:

- **Job Family:** A broad grouping of related roles (e.g., "Information Technology") used for logical organization and reporting.
- **Job Role:** A more specific definition within a family (e.g., "Software Engineer") that serves as the bridge between the Job Profile and the Job Classification.
- **Job Profile:** The descriptive container that holds the actual content of a role, including purpose, responsibilities, and required capabilities.
- **Job Classification (Job Code):** A foundation object that links the role to the corporate data model, defining structural attributes like Pay Grade, Employee Class, and Regular/Temporary status.
- **Position:** The specific instance of a job that exists within the organizational hierarchy. Positions are linked to Job Classifications (and thus Job Profiles) via the Job Code.

### 1.3.2. The Talent Intelligence Hub (TIH) Integration

In modern SuccessFactors environments, the Job Profile is further enriched by the Talent Intelligence Hub. This centralized framework manages workforce attributes across two primary repositories:

- **Attributes Library:** A repository for defining standardized skills, competencies, and behaviors. These attributes are mapped to Job Roles and surfaced in Job Profiles.
- **Growth Portfolio:** A person-centric repository where employees store their actualized and aspirational attributes. This allows the system to compare "Expected" proficiency (from the Job Profile) against "Actual" proficiency (from the Portfolio).

### 1.3.3. Key Technical Attributes

| Object | Key Attributes |
|---|---|
| Job Classification | Job Code, Job Title, Pay Grade, Employee Class, FLSA Status, Regular/Temporary |
| Job Profile | Purpose, Essential Duties, Skills (TIH), Competencies (TIH), Education, Certifications |
| Position | Position ID, Parent Position, Department, Business Unit, Job Code, FTE, Criticality |

## 1.3.4. Screenshots:

**Job Profile:**

.NET Developer
Role: .NET Developer Change

Job Description
The .NET Developer plays a key role in designing, developing, and optimizing scalable applications using the Microsoft .NET framework to fulfill organizational goals and enhance overall system performance. The position involves building secure, high-performance APIs and user interfaces, conducting comprehensive testing to ensure reliability, and delivering a seamless user experience across web, desktop, and mobile platforms. Strong command over C#, ASP.NET, and optionally F# or C++, combined with proficiency in database design and management, is essential for success in this role.

Job Role
.NET Developer

Job Family
Software Developers

Job Occupation
Software and Mathematics

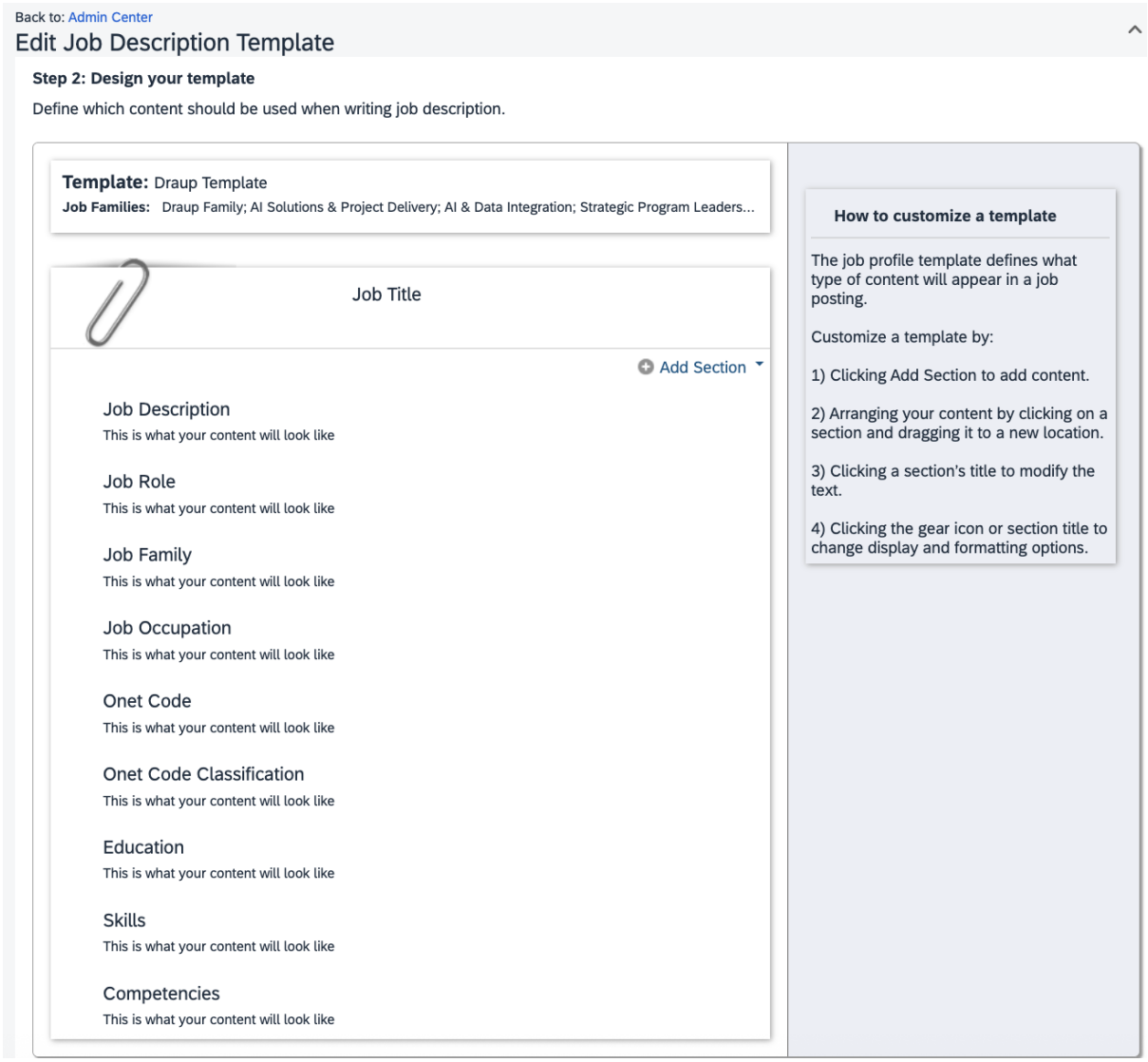
Onet Code
15-1252.00

Onet Code Classification
Software Developers

Education
PhD; Bachelors; Masters; Intermediate

Skills

## Job Profile Template:



**Skills in Job Profile:**

Skills

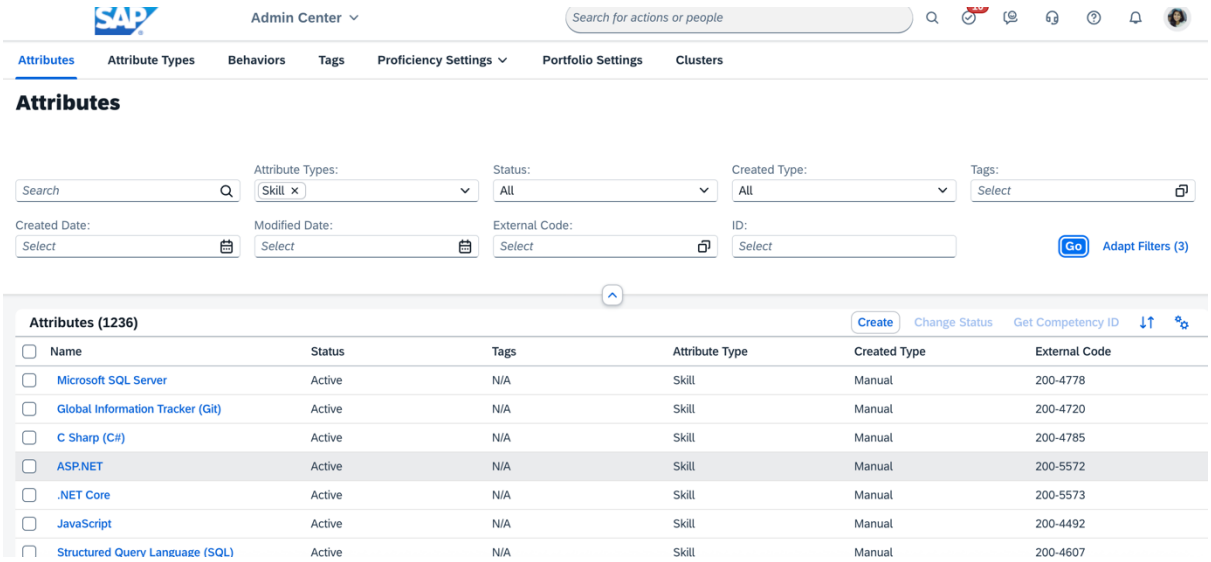
- PHP (Programming Language) ×
- AngularJS ×
- Microsoft SQL Server ×
- PostgreSQL ×
- CakePHP ×
- Node.js ×
- Spring MVC ×
- Struts ×
- Common Business Oriented Language (COBOL) ×
- C Sharp (C#) ×
- .NET Platform ×
- Structured Query Language (SQL) ×
- Global Information Tracker (Git) ×
- Version Control ×
- JavaScript ×
- ASP.NET ×
- Angular ×
- .NET Core ×

**Competencies in Job Profile:**

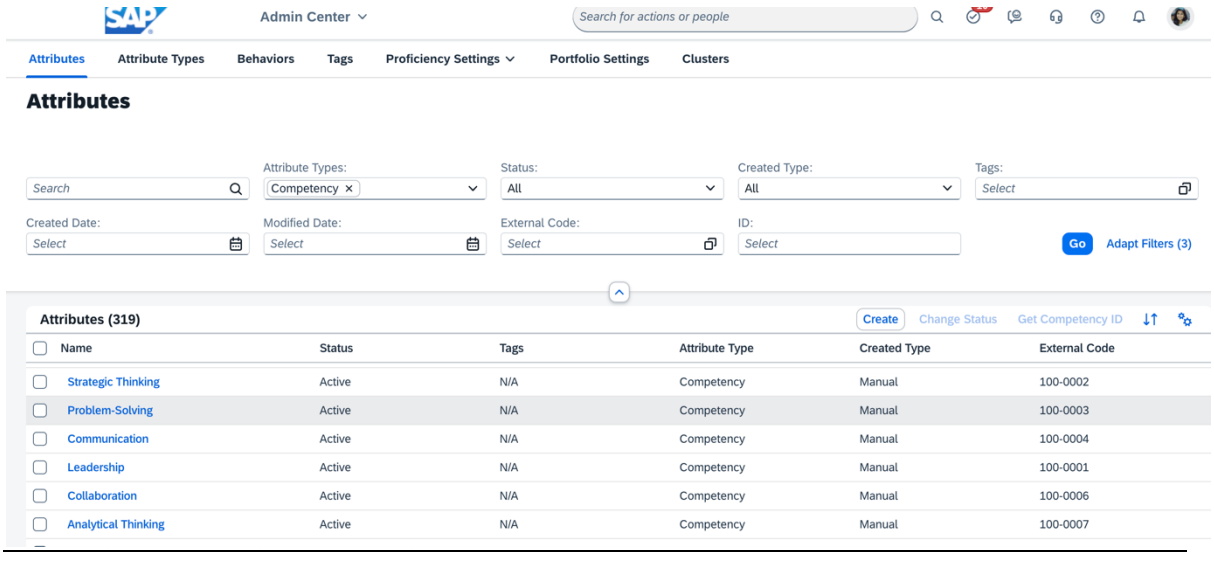
Competencies Add

- Developing Full Stack Solutions ×
- Developing Mobile Applications ×
- Cross-functional Collaboration ×
- Continuous Learning and Skill Development ×
- Project Management and Communication ×
- Developing Frontend Applications ×
- Technical Support and Training ×
- Custom Software Development ×
- Stakeholder Collaboration ×
- Database Management ×

## Talent Intelligence Hub Skills:




## Talent Intelligence Hub Competencies:



# 2. COMPONENT 2: SKILLS AND COMPETENCIES DATA OBJECT MODEL

While the job object provides the structural foundation, the skills and competencies objects are where the granular details of an employee's capabilities are stored. This component details the data models for skills in Oracle HCM, Workday, and SAP SuccessFactors, focusing on how they are stored, related, and attributed.

## 2.1. Oracle HCM Cloud: The Profile Management and Dynamic Skills Frameworks

In Oracle HCM Cloud, the relationship between Jobs, Skills, and Competencies is managed through Profile Management and the Dynamic Skills framework.

### 2.1.1. Object Model & Relationships

| Object | Level | Purpose | Relationship |
|---|---|---|---|
| Job | Workforce Structure | The administrative definition of a role. | Linked to a Model Profile (usually 1:1). |
| Model Profile | Talent | The "Requirement Blueprint" for a Job or Position. | Contains multiple Content Sections. |
| Content Section | Profile Component | A defined category (e.g., "Competencies" or "Skills"). | Configured to pull from a specific Content Subscriber. |
| Content Item | Library | The specific entity (e.g., "Leadership"). | Assigned to Model and Person Profiles. |

### 2.1.2. Skills vs. Competencies (The Two Frameworks)

**Oracle** maintains two distinct paths for talent data to allow for both structured behavioral assessments and fluid, AI-driven skill tracking.

**A. Job Competencies (Content Library)**

These are structured, behavioral traits traditionally managed within the Content Library.

- **Attributes:**
  - **Rating Model:** Defined proficiency levels (e.g., 1–5).
  - **Behaviors:** Specific indicators linked to proficiency levels to reduce bias.

    - **Evaluation:** Primarily updated via Performance Documents and manual Talent Review.

**B. Job Skills (Skills Nexus)**

These are technical or functional abilities managed via Oracle Dynamic Skills.

- **Attributes:**
    - **Skill Level:** Often uses a standard scale or AI-suggested levels.
    - **Skill Domain:** Categories like "Information Technology" or "Finance."
    - **AI Enrichment:** Automatically suggests synonyms (e.g., "Python" and "Python Scripting") and identifies "Trending Skills" in the market.

## 2.1.3. Simplified Architecture

**Level 1: The Catalogs (The Libraries)**

- **Content Library:** A validated list of behavioral competencies, degrees, and certifications.
- **Skills Nexus:** A dynamic, high-volume list of technical skills, often enriched by Oracle's AI data.
- **Rating Models:** The scales used to measure proficiency (e.g., Novice to Expert).

**Level 2: The Model Profile (The Requirement)**

The Model Profile is the "Template" that defines what is required for a Job.

- **Importance:** Allows HR to weight certain skills higher than others.
- **Target Level:** The minimum proficiency required for the role.
- **Criticality:** A flag used by Oracle Recruiting to filter or rank candidates based on fit.

**Level 3: The Person Profile (The Actual)**

The worker's specific record of their actual abilities.

- **Self-Rating:** Employee's assessment of their own proficiency.
- **Evaluated Rating:** Verified ratings from managers or through performance cycles.
- **Gap Analysis:** The system compares the Person Profile against the Job Model Profile to identify development needs.

### 2.1.4. Practical Workflow

1. **Define Structure:** You create a Job titled "Financial Analyst."
2. **Associate Profile:** You link it to a Job Model Profile.
3. **Add Competencies:** In the "Competencies" section, you pull "Analytical Thinking" from the Content Library and set a target level of "4 - Proficient."
4. **Add Skills:** In the "Skills" section, you pull "Data Visualization" from the Skills Nexus.
5. **Analyze Fit:** When a worker is assigned to the Job, the Profile Match tool calculates a percentage score by comparing the employee's Person Profile to these Job requirements.

## 2.2. Workday: The Dual Framework of Competencies and Skills Cloud

**Workday** utilizes two distinct but overlapping frameworks to manage human capital capabilities: the **Competency Model** and the **Skills Cloud**. While often used interchangeably, they function differently within the system's architecture, serving separate but complementary purposes.

### 2.2.1. The Competency Model (Structured & Evaluative)

The Competency Model is the traditional, manually curated framework used for formal performance management, compliance, and defining core job requirements.

- **Primary Object:** Competency
- **Structure:** Hierarchical and static. Competencies are defined by HR and tied to specific **Proficiency Scales** (e.g., 1–5 or Beginner–Expert).
- **Organization:** Grouped into **Competency Categories** (e.g., Leadership, Technical, Core Values) for clear organization.
- **Usage:** Used primarily in **Job Profiles** to set mandatory requirements and in **Performance Reviews** to formally measure worker behaviors against defined standards.

### 2.2.2. The Skills Cloud (Dynamic & AI-Driven)

The Skills Cloud is a modern, machine-learning ecosystem that treats skills as a fluid "language" across the enterprise. It is designed for agility, talent mobility, and workforce intelligence.

- **Primary Object:** Skill

- **Structure:** A massive, Workday-maintained ontology of over 55,000 skills. The system automatically identifies synonyms (e.g., mapping "Data Analysis" to "Analytics") to create a clean, unified taxonomy.
- **Inference:** Unlike competencies, skills can be **Inferred** by Workday's AI based on a worker's resume, job history, project experience, or performance feedback, providing a more holistic view of their capabilities.
- **Usage:** Powers the **Talent Marketplace**, suggesting internal gigs, projects, and career paths by matching **Worker Skills** to the needs of different roles and opportunities.

### 2.2.3. Data Object Relationship Mapping

| Business Need | Object A | Relationship | Object B |
|---|---|---|---|
| Job Requirement | Job Profile | Defines needs via | Competency / Skill |
| Worker Talent | Worker | Possesses | Worker Skill |
| Evaluation | Worker | Rated against | Proficiency Scale |
| Categorization | Skill | Belongs to | Skill Category |

### 2.2.4. Key Differences at a Glance

| Feature | Competency Model | Skills Cloud |
|---|---|---|
| Creation | Manual (HR Defined) | Dynamic (Workday Library) |
| Measurement | Proficiency Scales (1–5) | Binary or Inferred Leveling |
| Update Frequency | Periodic (Annual/Semi-Annual) | Real-time / Continuous |
| Main Use Case | Compliance & Performance Reviews | Career Mobility & Talent Marketplace |

### 2.2.5. Summary for Implementation

- **Competencies** are best suited for high-stakes, formal requirements that require a specific, measurable grade or score for compliance and performance management.

- **Skills** are ideal for high-volume talent matching, identifying hidden talent within the organization, and mapping the general capabilities of the workforce for strategic planning and internal mobility.
- **Integration:** Both Competency and Skill objects are linked to the Worker and Job Profile objects, allowing for a comprehensive gap analysis between what a job requires and what a worker offers, combining both formal requirements and dynamic capabilities.

## 2.3. SAP SuccessFactors: The Talent Intelligence Hub

The semantic model for skills in SuccessFactors has evolved from a static library into a dynamic **Intelligence Hub.** It treats capabilities as living data points that connect the individual's potential to the organization's needs.

### 2.3.1. The Capability Taxonomy

The model categorizes human capability into two distinct semantic types, unified under a single "Attribute" framework:

- **Competencies:** These are modeled as observable behaviors and foundational strengths (e.g., "Strategic Thinking" or "Communication"). They are typically linked to performance standards and are measured against defined proficiency scales.
- **Skills:** These are modeled as specific technical or functional abilities (e.g., "Data Analysis" or "Python Programming"). While competencies are often stable over a career, skills are treated as highly dynamic, with shorter lifecycles.

### 2.3.2. The "Whole Self" Model

The data model for an employee's capability is divided into three semantic perspectives:

1. **Requirement Data (The Demand):** Attributes mapped directly to the Job Role. This defines what the organization *needs* for a specific position to be successful.
2. **Actualized Data (The Supply):** The verified skills and competencies an employee has demonstrated, often validated through assessments or manager feedback.
3. **Aspirational Data (The Gap):** A unique layer that stores "interests." This allows the model to capture not just what an employee *can* do, but what they *want* to do, enabling predictive talent matching.

### 2.3.3. Proficiency and Normalization

To ensure data can be compared across a global enterprise, the model uses **Normalized Proficiency Scales.** This allows the system to semantically understand that a "Level 4" in a technical skill for an Engineer is equivalent in weight to a "Level 4" for a Designer, even if the specific tasks differ. This normalization is what allows for sophisticated gap analysis and internal mobility matching.

# 3. COMPONENT 3: HCM INTEGRATION TOOLS AND APIs

The true power of an HCM system is realized not just through its internal data model, but through its ability to integrate with the broader enterprise technology landscape. This component details the integration tools and APIs provided by each of the two HCM vendors, with a focus on how they enable the import and export of skill data and facilitate enhancements from external skill intelligence vendors.

## 3.1. Oracle HCM Cloud: A Suite of Integration Options

Oracle provides a comprehensive suite of integration tools, catering to different use cases from real-time API calls to bulk data loading.

### 3.1.1. Key Integration Tools:

| Tool | Purpose | Use Case |
|---|---|---|
| REST API | Real-time data access and management | Synchronous integrations, UI-driven interactions. |
| HCM Data Loader (HDL) | Bulk data import and export | Initial data migration, mass updates, large-scale sync. |
| HCM Extracts | Data extraction for reporting and analytics | Feeding data to downstream systems, analytics platforms. |
| Oracle Integration Cloud (OIC) | Pre-built adapters and orchestration | Complex, multi-step integration workflows. |
| SOAP Web Services | Legacy integration interface | Integration with older systems that do not support REST. |

### 3.1.2. API and Data Enrichment Capabilities:

For integrating with external skill intelligence vendors like **Draup**, Oracle HCM provides several key capabilities:

- **Dynamic Skills API Access**: REST endpoints are available for accessing and managing skills data, including AI-powered skills recommendations and matching.
- **Profile Management Integration**: The HRT_PROFILE_ITEMS table, which stores skills, is accessible via APIs, allowing for the programmatic update of skills, competencies, and proficiency levels.
- **Bulk Data Operations**: HDL can be used for mass updates of skills data, enabling the import of enriched skill taxonomies from external vendors.

## 3.2. Workday: The Integration Cloud Platform

Workday's integration strategy is centered around its **Integration Cloud Platform**, which provides a suite of tools for building, deploying, and managing integrations without the need for on-premises middleware.

### 3.2.1. Key Integration Tools:

| Tool | Purpose | Use Case |
|---|---|---|
| Workday Studio | Advanced, graphical integration development | Complex, custom integrations with extensive logic. |
| Enterprise Interface Builder (EIB) | Simple, template-driven data import/export | Business-user-friendly data loading and extraction. |
| Workday Integration Cloud Connect | Pre-built connectors to third-party systems | Rapid integration with common applications (e.g., Salesforce). |
| Web Services (SOAP & REST) | Programmatic access to Workday business objects | Custom application development, real-time data exchange. |
| Report-as-a-Service (RaaS) | Exposing Workday reports as web services | Data extraction for analytics and downstream systems. |

### 3.2.2. API and Data Enrichment Capabilities:

Workday's skills import/export feature and its partnership with skills intelligence vendors like **Draup** are key to its data enrichment strategy:

- **Pre-Mapped Ontologies**: Workday works with partners to provide pre-mapped skill ontologies, simplifying the process of importing and normalizing skills from external sources.
- **Skills Cloud API**: The Skills Cloud provides APIs for accessing and managing skills data, including proficiency levels and endorsements.

- **EIB for Skills**: The Enterprise Interface Builder can be used to perform bulk imports and exports of skills data, facilitating large-scale enrichment projects.

## 3.3. SAP SuccessFactors: Integration Center and OData APIs

SAP SuccessFactors provides a comprehensive suite of integration tools, including its **Integration Center** and a rich set of **OData APIs**. These tools enable seamless data exchange between SuccessFactors and external systems, including skill intelligence platforms like **Draup**.

### 3.3.1. Key Integration Tools:

| Tool | Purpose | Use Case |
|---|---|---|
| OData V2 API | Real-time data access and management | Synchronous integrations, external application access. |
| Integration Center | Pre-built integration templates and workflows | Scheduled data imports/exports, third-party integrations. |
| Intelligent Services | AI-powered data enrichment and recommendations | Skills inference, career path suggestions. |
| Data Shares | Secure data sharing with external partners | Sharing workforce data with analytics platforms. |

### 3.3.2. API and Data Enrichment Capabilities:

For integrating with external skill intelligence vendors like **Draup**, SAP SuccessFactors provides several key capabilities:

- **Talent Intelligence Hub API**: Access to the centralized Attributes Library for skills and competencies, enabling external vendors to enrich and normalize skill data.
- **Growth Portfolio Integration**: External systems can read and write to the employee Growth Portfolio, adding inferred skills or updating proficiency levels based on external assessments.
- **Bulk Data Operations**: The Integration Center supports scheduled imports and exports of skills data, facilitating large-scale enrichment projects with external vendors.

# 4. COMPONENT 4: SKILL CONFLICT, TAXONOMY, ANDF NOMENCLATURE MANAGEMENT

The most significant challenge in creating a unified skills ecosystem is the management of the skills taxonomy itself. Different systems, departments, and individuals often use conflicting terminology for the same skill, leading to a fragmented and unreliable skills inventory. This component examines how Oracle HCM, Workday, and SAP SuccessFactors handle skill naming conventions, taxonomy conflicts, and nomenclature standardization.

## 4.1. The Core Challenge: Semantic Inconsistency

At its heart, the problem is one of semantics. A skill like "JavaScript" might be referred to as "JS," "ECMAScript," or "JS Programming." Without a mechanism to normalize these terms, an organization cannot get an accurate picture of its capabilities. Both HCM vendors have developed strategies to address this, centered around AI, partnership, and flexible data models.

## 4.2. Oracle HCM Cloud: AI-Powered Normalization

Oracle's strategy is heavily reliant on its **Dynamic Skills** engine, which uses AI and machine learning to automate the process of skill normalization and taxonomy management.

**Conflict Resolution Mechanisms:**

- **AI-Powered Skills Inference**: The system automatically scans unstructured text in job descriptions, resumes, and performance reviews to identify and suggest skills. It uses natural language processing (NLP) to understand the context and map variants to a canonical skill.
- **Oracle-Seeded Skills Library**: Oracle provides a vast, curated library of skills that serves as a baseline taxonomy. This library is continuously updated with emerging skills and industry-specific terminology.
- **Skill Normalization**: When different terms for the same skill are encountered, the system maps them to a single, canonical term in the skills library, while retaining the original terms as aliases for search purposes.

## 4.3. Workday: Partner-Driven Ontology Mapping

Workday's approach emphasizes its ecosystem of partners and a new skills import/export feature designed to create a "bi-directional flow of skills."

**Conflict Resolution Mechanisms:**

- **Pre-Mapped Skill Ontologies**: Workday collaborates with skills intelligence vendors to create and maintain pre-mapped ontologies. This means that when skills are imported from these partner systems, they are automatically mapped to the corresponding skill in Workday's taxonomy.
- **Unified Skill Definitions**: The goal of the skills import/export feature is to simplify conflicting terminology and create uniform skill definitions across the enterprise, regardless of the source system.
- **Skills Ecosystem Integration**: By fostering a strong partner ecosystem, Workday offloads much of the complexity of taxonomy mapping to specialized vendors, allowing customers to leverage their expertise.

## 4.4. SAP SuccessFactors: Talent Intelligence Hub and Normalized Attributes

SAP SuccessFactors leverages its **Talent Intelligence Hub** to manage skill taxonomies and resolve conflicts. The **Attributes Library** serves as the central repository for all skills and competencies, providing a single source of truth for the organization.

**Conflict Resolution Mechanisms:**

- **Centralized Attributes Library**: All skills and competencies are stored in the Attributes Library, which acts as the master catalog. This centralization ensures that the same skill is represented consistently across the entire platform, regardless of its source.
- **Normalized Proficiency Scales**: The use of normalized proficiency scales allows the system to compare skills across different roles and geographies, ensuring that a "Level 4" in one context is equivalent to a "Level 4" in another.
- **Integration with External Vendors**: SAP SuccessFactors integrates with external skill intelligence vendors like **Draup** to enrich the Attributes Library with market-validated skill definitions and emerging skills, helping to keep the taxonomy current and relevant.

# 5. COMPONENT 5: THE UNIFIED SKILLS ECOSYSTEM

In a large-scale enterprise, a successful skills strategy relies on a simple "Hub and Spoke" relationship between the core HCM, specialized Intelligence Vendors, and internal execution tools. This model provides a clear framework for data flow, ownership, and governance.

### 1. The Core: HCM (Workday, Oracle, or SAP SuccessFactors)

The HCM is the **System of Record**. It provides the official structure for the organization and is the final destination for all verified employee and job data.

- **Key Function:** The HCM links skills to official business actions like promotions, staffing, compensation, and performance management.
- **The Goal:** To serve as the "Single Source of Truth" for the workforce's official capabilities, ensuring that all talent decisions are based on a consistent and reliable dataset.

### 2. The Intelligence: Draup

Platforms like **Draup** provide the **Market Intelligence Layer**. They act as an "outside-in" lens to keep the internal skills data relevant to the global labor market, preventing the skills taxonomy from becoming stale or disconnected from real-world trends.

- **Key Function:** Draup identifies emerging "Sunrise" skills and declining "Sunset" skills, maps the internal talent pool against industry benchmarks, and enriches employee profiles with skills inferred from external data sources.
- **The Goal:** To automate the discovery of skills that employees may not have manually added to their profiles and to provide a predictive view of the skills the organization will need in the future.

### 3. The Evidence: Internal Systems

These are the **Execution Tools** where actual work happens, such as Jira, Salesforce, or GitHub. They provide the "digital proof" of a skill through real-world project delivery and output.

- **Key Function:** These systems provide concrete evidence that a skill is not just claimed, but is being actively applied to drive business value. This data can be used to validate skill proficiency and identify subject matter experts.
- **The Goal:** To ground the skills data in tangible business outcomes, moving beyond self-reported proficiency to a more objective, evidence-based assessment of capabilities.

## CONCLUSION: FROM SKILLS DATA TO WORKFORCE DECISIONS

The transition from job-centric to skills-centric talent strategies is often framed as a cultural or organizational shift. In practice, it is far more concrete. It is a question of data models, system boundaries, integration choices, and governance discipline.

As this paper has shown, Oracle HCM Cloud, Workday, and SAP SuccessFactors each offer robust—but distinct—approaches to representing jobs, skills, and competencies. Oracle emphasizes profile-driven requirements enriched by AI-powered Dynamic Skills. Workday integrates skills deeply into an object-oriented, in-memory architecture designed for mobility and inference. SAP SuccessFactors separates roles from positions and centralizes skills through the Talent Intelligence Hub, enabling a more flexible, person-centric view of capability.

None of these platforms, on their own, fully solves the enterprise skills challenge.

**The future of skills-based talent management lies in how these systems are connected, not how they are configured in isolation.** A unified skills ecosystem—where the HCM serves as the authoritative system of record, external intelligence platforms continuously enrich and normalize skills, and internal execution systems provide evidence of skills in action—creates the conditions for better decisions.

For Boards and executive leadership teams, the implications are practical and immediate:

- **Skills strategy must be treated as enterprise infrastructure**, not an HR initiative
- **Taxonomy and normalization require ongoing intelligence,** not one-time design
- **Internal skills data must be continuously validated against market reality**
- **Workforce decisions improve when skills are tied to evidence, not self-reporting**

For CHROs, this reframes the role of the function—from maintaining talent systems to orchestrating a skills ecosystem that enables agility, credibility, and speed.

Enterprises that approach skills transformation deliberately—grounded in architecture, integration, and evidence—will be better positioned to redeploy talent, absorb disruption, and align workforce capability with business strategy. Those that do not risk building skills programs that are rich in aspiration, but fragile in execution.

The opportunity is not simply to manage skills better, but to make workforce decisions with greater confidence, consistency, and foresight. That is the real promise of a unified enterprise skills ecosystem.