Comprehensive Feasibility Analysis and Migration Protocol: Transitioning from PTC Arena to Open Source PLM Architectures

Executive Summary

The strategic imperative to migrate from a proprietary, cloud-native Software-as-a-Service (SaaS) platform like PTC Arena to an open-source Product Lifecycle Management (PLM) solution is typically driven by a desire for data sovereignty, licensing cost reduction, and architectural flexibility. However, executing such a migration requires navigating a complex landscape of data dependency, proprietary formats, and varying definitions of "open source" within the industrial software market. This report provides an exhaustive feasibility analysis, tool selection justification, and a granular technical execution guide for migrating from PTC Arena to an open-source alternative.

The analysis confirms that the migration is **technically feasible**, primarily due to PTC Arena's adherence to the IPC-2570 (Product Data eXchange or PDX) standard for data export.[1] This standard provides a structured, XML-based vector for extracting intelligent product data, which mitigates the "data gravity" issues typically associated with multi-tenant SaaS platforms. However, feasibility is not binary; while structural data (BOMs, Parts) and unstructured files (CAD, PDFs) can be migrated, historical transaction logs (audit trails), proprietary workflow configurations, and integrated supply chain intelligence (e.g., SiliconExpert connectors) generally cannot be transferred with full fidelity.[3]

Following a rigorous comparative analysis of the current open-source PLM landscape—specifically examining Aras Innovator, Odoo, and DocDokuPLM—this report selects **DocDokuPLM** as the optimal target system. While Aras Innovator offers superior enterprise depth, its 2024/2025 licensing restructuring has effectively removed it from the "free/open" category for production environments exceeding 50 users.[5] Odoo's Community Edition lacks native PLM functionality.[8] DocDokuPLM remains the only viable, fully

open-source (AGPL), cloud-native solution capable of replicating Arena's document-centric architecture without artificial user caps or licensing fees.[9]

The proposed migration methodology utilizes a hybrid extraction approach: leveraging PDX packages for high-fidelity metadata and BOM structure extraction, combined with custom Python-based REST API scripting for bulk file attachment retrieval. The transformation logic addresses critical disparities between Arena's flat revisioning (numeric/alpha) and standard version branching, ensuring data integrity is preserved in the target system.

---

1. Feasibility Study: The Architecture of Extraction

The feasibility of migrating away from PTC Arena acts as a function of the source system's data accessibility and the target system's ability to ingest complex, relational product structures. Unlike on-premise SQL-based systems where data extraction might involve direct database queries, Arena operates as a multi-tenant cloud service. Access is strictly mediated through the Application Layer via User Interface (UI) exports or Application Programming Interfaces (APIs).

1.1 Data Gravity and SaaS Lock-In Mechanisms

In the context of cloud PLM, "data gravity" refers to the accumulation of data and logic that makes migration increasingly difficult over time. Arena utilizes a proprietary schema optimized for multi-tenancy. The primary challenge in migration is not accessing the data, but disentangling the *content* (Parts, BOMs, Documents) from the *context* (Workflow history, Discussion threads, Redlines).

The analysis identifies three distinct classes of data residing within Arena, each presenting different feasibility profiles for migration:

1. **Structured Product Data (High Feasibility):** This includes Item Masters, Bills of Materials (BOMs), Approved Manufacturer Lists (AML), and Vendor definitions. Arena supports the export of this data via CSV (Comma Separated Values) and PDX (Product Data eXchange).[1] The PDX format is critical here; unlike CSV, which flattens hierarchical relationships, PDX preserves the parent-child structure of complex assemblies, making it the preferred vehicle for BOM migration.[11]
2. **Unstructured Content (Moderate Feasibility):** This category encompasses CAD files, datasheets, specifications, and other binary attachments linked to items. While Arena

allows for bulk download via the UI, transferring terabytes of data reliably requires programmatic intervention using the REST API.[12] The feasibility is constrained by bandwidth, API rate limits, and the complexity of re-linking extracted files to their correct item revisions in the target system.

3. **Process and Behavioral Data (Low Feasibility):** This includes the "digital thread" of engineering change orders (ECOs), sign-off histories, user comments, and redline markups. While the *result* of a change (e.g., Rev A moving to Rev B) is migratable, the *transactional evidence* (e.g., "John Doe signed off on 12/12/2024") is often lost or must be archived as a static report (PDF) rather than active database entries.[4]

1.2 The Role of IPC-2570 (PDX) in Migration Feasibility

The existence of the IPC-2570 standard is the single most significant enabler of this migration. Developed by the International Electronics Manufacturing Initiative (iNEMI), PDX was designed specifically to facilitate data exchange between Original Equipment Manufacturers (OEMs) and Electronics Manufacturing Services (EMS) providers.[2]

Because Arena uses PDX as a native export format, the migration team does not need to reverse-engineer the proprietary database schema. The PDX package (a ZIP file containing pdx.xml and attachments) provides a standardized, XML-based representation of the product record. This XML schema defines:

- **Items:** The core attributes of parts and documents.
- **BillOfMaterial:** The relational links defining assembly structures.
- **ApprovedManufacturerList:** The sourcing relationships between internal part numbers and manufacturer part numbers.
- **Changes:** The header information for Engineering Change Requests (ECRs) and Orders (ECOs).[15]

The feasibility of the migration rests on the ability to parse this XML structure and map it to the target system's data model. Since PDX is an open standard, parsers are readily available or can be constructed using standard XML libraries (e.g., Python's lxml), reducing the technical risk significantly compared to migrating from a system with binary or obfuscated export formats.

1.3 API Accessibility and Constraints

For data elements not fully covered by PDX—specifically large file repositories and granular metadata not included in standard export packages—Arena's REST API provides the necessary bridge. The API follows a standard resource-oriented architecture using JSON payloads and OAuth 2.0 authentication.[13]

However, reliance on the API introduces latency. Unlike a database dump which takes minutes, iterating through 100,000 items via API GET requests can take days. The feasibility assessment must account for:

- **Throttling:** Arena imposes limits on API calls per minute/hour.
- **Pagination:** Scripts must handle paginated responses for large datasets (e.g., returning 100 items at a time).
- **Error Handling:** Transient network failures during file downloads require robust retry logic.

1.4 Feasibility Conclusion

The migration is feasible. The combination of PDX for structural integrity and REST APIs for binary extraction covers approximately 90-95% of the business-critical data required to operate a PLM system. The remaining 5-10% (audit logs, deep workflow history) represents a trade-off inherent in moving between disparate platforms.

---

2. Market Analysis and Target Tool Selection

The term "Open Source" in the PLM market is often used loosely, covering a spectrum from "Community Editions" of proprietary software to true Free and Open Source Software (FOSS). Selecting the correct target is critical to avoiding a "false migration" where the organization moves from one vendor lock-in to another.

2.1 The "Open Source" Mirage: Aras Innovator

Aras Innovator has long been the dominant player in the "open" PLM space. Historically, Aras offered a "Enterprise Open Source" model where the software was free to download and use,

with revenue generated solely through subscription services for support and upgrades.

However, the landscape for Aras changed significantly in the 2024/2025 release cycles.

- **Licensing Shift:** The Aras Community Edition is now explicitly marketed for "Evaluation and Proof of Concept" or "Development Environments".[5]
- **Usage Restrictions:** Recent license agreements and release notes indicate a hard or soft cap of **50 named users** for the Community Edition.[17] Furthermore, the software is provided "as-is" with restricted access to the subscriber portal, making security patching difficult for production environments.
- **Production Viability:** While technically capable, running a mission-critical PLM for a mid-to-large enterprise on the Aras Community Edition violates the spirit (and potentially the letter, depending on interpretation of "commercial use") of their new terms. The risk of a licensing audit or lack of upgrade path disqualifies it for an organization seeking a strictly open-source, unrestricted future.[6]

2.2 The ERP Dependency: Odoo PLM

Odoo is a popular open-source ERP platform. However, its PLM capabilities are segmented.

- **Community Edition (Free):** The free version of Odoo includes the MRP (Manufacturing Resource Planning) module but **excludes** the dedicated PLM module.[8] Key features like Engineering Change Orders (ECOs), version control for BOMs, and difference comparisons are gated behind the Enterprise (paid) license.[20]
- **Functional Gap:** Attempting to use Odoo Community as a PLM requires significant custom development to build version control on top of the basic MRP tables. This defeats the purpose of adopting a specialized tool.

2.3 The True Open Source Contender: DocDokuPLM

DocDokuPLM stands out as a genuine open-source solution licensed under the **GNU Affero General Public License v3.0 (AGPL)**.

- **Architecture:** It is a cloud-native application built on a modern Java stack (Java EE/Jakarta EE), utilizing a Payara (GlassFish) application server, and leveraging standard databases (MongoDB/PostgreSQL) and search engines (ElasticSearch).[9]
- **Feature Parity:** DocDokuPLM natively supports:
    - **Document Management:** Strong version control, check-in/check-out, and

visualization, mirroring Arena's document-centric approach.[10]
    - **Product Structure:** Part-to-Part relationships (BOMs) with effectivity dates.
    - **Visualization:** WebGL-based 3D visualization in the browser, a feature often lacking in legacy open-source tools.[10]
- **Licensing:** There are no user limits, no feature gates, and no "non-production" clauses. The AGPL ensures the code remains open.

2.4 Selection Verdict

DocDokuPLM is selected as the target tool.

- *Why not Aras?* The 50-user cap and non-production licensing terms create unacceptable legal and operational risk for a migration intended to escape vendor constraints.
- *Why not Odoo?* The Community Edition lacks PLM functionality, and the Enterprise Edition is not open source.
- *Why DocDoku?* It offers the closest architectural alignment (web-based, document-centric) to Arena and is the only thoroughly featured PLM available under a truly free license (AGPL) suitable for production use without user caps.

---

3. Target System Architecture: DocDokuPLM

Understanding the target architecture is a prerequisite for mapping the data. DocDokuPLM operates on a containerized microservices model, which differs from the monolithic database structure often assumed in legacy migrations.

3.1 Technology Stack

The DocDokuPLM stack consists of the following components, all of which must be provisioned for the migration [22]:

- **Application Server:** Payara Server (a derivative of GlassFish), hosting the Java-based business logic.
- **Database (Metadata):** MongoDB or PostgreSQL. The NoSQL nature of MongoDB (if selected) aligns well with the hierarchical, variable-attribute nature of PLM data, similar

to the flexible schema of Arena.

- **Search Index:** ElasticSearch. This powers the search bar and facet filtering. Migration scripts must trigger re-indexing after bulk loads to ensure data is discoverable.
- **File Storage:** A standard file system or S3-compatible object storage. This is where the binary files extracted from Arena will reside.

3.2 Data Model Comparison

Concept	PTC Arena Terminology	DocDokuPLM Terminology	Mapping Strategy
Object Identity	Item Number (e.g., 030-0001)	Part Number / Document ID	Direct mapping. Ensure uniqueness constraints in DocDoku.
Versioning	Revision (Numeric 01 or Alpha A)	Version (Major) + Iteration (Minor)	Map Arena Revision to DocDoku Version. Iteration typically starts at 1.
State	Lifecycle Phase (e.g., In Production)	Lifecycle State (e.g., RELEASED)	Create a mapping table. Arena "Obsolete" maps to DocDoku "Obsolete".
Structure	BOM (Parent-Child Link)	Part Link / Composition	Recursive relationship. Quantity and Unit of Measure (UOM) must be mapped.
Files	Attachments (File on Item)	Document (Object) + Attached File	Arena files become "Document" objects in DocDoku, which

			are then linked to the "Part".
Suppliers	Vendor / Manufacturer Part	Partner / Supplier Part	Suppliers are distinct organizations in DocDoku.

---

4. Migration Methodology: Detailed Execution Steps

The migration is executed in four sequential phases: Preparation, Extraction, Transformation, and Loading (ETL).

Phase 1: Preparation and Environment Provisioning

Before touching the source data, the target infrastructure must be established and the data scope defined.

Step 1.1: Deploying DocDokuPLM

Using Docker is the most efficient method to stand up the target environment.

1. **Clone the Docker Repository:**
   Bash
   ```
   git clone https://github.com/docdoku/docdoku-plm-docker.git
   cd docdoku-plm-docker
   ```

2. Infrastructure Tuning:
   PLM systems are memory-intensive. Ensure the host machine has at least 16GB RAM. The ElasticSearch component requires specific kernel parameter adjustments 22:
   Bash
   ```
   sysctl -w vm.max_map_count=262144
   ```

3. Configuration:
   Modify the env/back.env file to set the DOCDOKU_PLM_CODEBASE URL and database credentials.
4. Launch:
   Execute docker-compose up -d. Verify access at http://localhost:8080.

Step 1.2: Workspace Configuration

In the DocDoku admin console, create a specific workspace (e.g., ARENA_MIGRATION). This isolates the migrated data from any test data. Configure the *Attributes* and *Templates* in DocDoku to match the custom fields defined in Arena (e.g., "RoHS Compliance", "Project Code").[23]

Phase 2: Data Extraction (The Source)

This phase focuses on getting data out of Arena. We will use a hybrid approach: PDX for metadata and API for files.

Step 2.1: The PDX Export (Metadata & BOMs)

The PDX package is the "source of truth" for the product structure.

1. **Trigger Export:** In Arena, navigate to the **Items** view. Select the top-level assemblies or "All Items."
2. **Export Settings:** Choose **Actions > Export > PDX**.
   - **Filter:** Select "Latest Released" to export only the current active configuration. Migrating full revision history via PDX creates massive complexity and is often not recommended for open-source targets.[15]
   - **Content:** Ensure "Items," "BOMs," "Manufacturers," and "Item Files" (metadata) are checked.
   - **Whitespace Cleaning:** Enable options to remove trailing whitespace and normalize line endings (CRLF), as described in [15], to prevent parsing errors later.
3. **Result:** A .pdx file (ZIP archive) containing pdx.xml.

Step 2.2: The API Extraction (File Attachments)

While PDX lists the files, extracting gigabytes of attachments via PDX is often slow or hits size limits. A direct API script is more robust.

Python Script Logic for File Extraction:

1. Authentication:
   Authenticate with the Arena REST API using the admin credentials to obtain a JSESSIONID or OAuth token.13
2. Parsing the Manifest:
   Read the pdx.xml file (from Step 2.1) using the lxml library. Extract a list of all Item GUIDs and their associated File GUIDs.
3. Download Loop:
   Iterate through the list. For each file:
   - Endpoint: GET /items/{itemGuid}/files/{fileGuid}/content.[12]
   - **Rate Limiting:** Implement a time.sleep(1) between requests to respect Arena's API throttling.
   - **Storage:** Save the file to a local directory structure, e.g., ./staging/files/{ItemNumber}/{Revision}/{Filename}.
   - **Validation:** Compare the downloaded file size against the FileSize attribute in the PDX XML to ensure incomplete downloads are detected and retried.

Phase 3: Data Transformation (The Logic)

This is the most critical phase. Raw Arena data does not fit directly into DocDoku.

Step 3.1: XML Parsing and Object Construction

Using Python, parse the pdx.xml to build JSON payloads for DocDoku.

- **Lifecycle Mapping:** Arena might use "In Production" while DocDoku expects "RELEASED". A mapping dictionary is required:
  Python
  lifecycle_map = {

```
  "In Production": "RELEASED",
  "Obsolete": "OBSOLETE",
  "In Design": "IN_WORK"
}
```

- **Attribute Mapping:** Map Arena's custom attributes (e.g., attribute_123) to DocDoku's property fields. If DocDoku uses strict typing, ensure text fields in Arena containing numbers are converted to the correct data type.

Step 3.2: Revision Normalization

Arena allows arbitrary revision labels (e.g., "Pre-Prod 1"). DocDoku enforces a strict Version/Iteration schema.

- **Strategy:** Map Arena revisions to a sequence.
  - Arena "1" -> DocDoku Version 1, Iteration 1.
  - Arena "A" -> DocDoku Version A, Iteration 1.
- **Caution:** If migrating history, you must sort the revisions chronologically in the extraction phase to load them in the correct order.

Step 3.3: BOM Hierarchy Reconstruction

The PDX format defines BOMs as a list of BillOfMaterial elements.

- **Logic:**
  1. Create a dictionary of all Items.
  2. For each BillOfMaterial entry, identify the Parent and the Child.
  3. Construct a "Link" payload for DocDoku that specifies the Child's Part Number, Version, Quantity, and Reference Designators (RefDes).[25]

Phase 4: Data Loading (The Target)

Data loading utilizes the DocDoku REST API.[26] Direct database injection is discouraged as it bypasses business logic validation.

Step 4.1: Organization and User Seeding

Before loading parts, ensure the "Owner" of the parts exists.

- **Action:** Create user accounts in DocDoku corresponding to the "Created By" users in Arena. If those users are no longer with the company, map them to a generic "migration_admin" user.

Step 4.2: Importing Parts (Leaf Nodes First)

To avoid dependency errors, import items in a specific order:

1. **Leaf Nodes:** Parts that have no children (e.g., Resistors, Capacitors, Screws).
2. **Sub-Assemblies:** Parts that contain Leaf Nodes.
3. **Top-Level Assemblies:** Final products.

API Call:

- POST /api/workspaces/{workspaceId}/parts
- Payload: JSON object created in Phase 3.

Step 4.3: Uploading and Linking Documents

This step links the files downloaded in Phase 2 to the parts created in Step 4.2.

1. **Upload Binary:**
   - Use POST /api/workspaces/{workspaceId}/files with multipart/form-data.[26]
   - Capture the returned uuid or id.
2. **Create Document Meta-Object:**
   - Create a "Document" entity in DocDoku containing the file title and revision.
3. **Link to Part:**
   - Use the Part-Document link endpoint to attach the Document to the specific Part Revision.
   - *Insight:* In Arena, files are often attached directly to the Item. In DocDoku, it is best practice to manage them as "Document" objects linked to the Part, preserving the distinction between the physical part and its specification.

Step 4.4: BOM Structure Linking

Once all parts exist, create the BOM links.

- **Loop:** Iterate through the BOM definitions.
- **API Call:** POST /api/workspaces/{workspaceId}/parts/{parentID}/links
- **Payload:** Includes childID, quantity, unit.
- **Error Handling:** If a child part is missing (e.g., it was filtered out during export), the API will return a 404. Log these errors to a missing_components.csv file for manual review.

Phase 5: Verification and Validation

A successful API response does not guarantee data integrity.

Step 5.1: Quantitative Validation

- **Count Verification:** Compare the total number of Active Items in Arena vs. DocDoku.
- **File Checksums:** Calculate the MD5/SHA-256 hash of the files on the DocDoku server and compare them to the source files from Arena to ensure no corruption occurred during transfer.[27]

Step 5.2: BOM Integrity Check

Perform a "Where-Used" and "Composition" test on a random sample of 5-10 complex assemblies.

- **Visual Compare:** Open the BOM in Arena and DocDoku side-by-side. Check for:
  - Correct Quantities.
  - Correct Units of Measure (e.g., did "inches" convert to "mm"?).
  - Reference Designators (e.g., R1, C4) – critical for electronics.[28]

5. Operational Sustainability and TCO Analysis

Migrating from a SaaS model (Arena) to an open-source model (DocDoku) radically shifts the cost structure and operational responsibility.

5.1 The Hidden Cost of "Free"

While DocDoku incurs no licensing fees, the Total Cost of Ownership (TCO) is not zero.

- **Infrastructure:** You must pay for the cloud compute (AWS/Azure) or on-premise servers to host the Docker containers.
- **Personnel:** Arena includes maintenance, backups, and security patching in the subscription. With DocDoku, these become the responsibility of internal IT. A dedicated System Administrator or DevOps engineer is required to manage the Payara server, database backups, and SSL certificates.[29]

5.2 Loss of Integrated Supply Chain Intelligence

One of Arena's strongest features is its integration with component databases (e.g., Octopart, SiliconExpert) to flag obsolete parts or compliance risks (RoHS/REACH).[30]

- **Gap Analysis:** DocDoku does not have these pre-built connectors. The organization loses automated "End of Life" (EOL) alerts.
- **Mitigation:** Component engineers will need to manually check part status or the development team must build custom scrapers/connectors to pull data from third-party APIs into DocDoku attributes.

5.3 Compliance and Validation (FDA/ISO)

If the organization operates in a regulated industry (Medical Devices, Automotive, Aerospace):

- **Arena:** Comes with a validation package and is Part 11 compliant (electronic signatures, audit trails).[3]
- **DocDoku:** While it supports versioning, it is not "pre-validated." The organization must write and execute its own Installation Qualification (IQ), Operational Qualification (OQ), and Performance Qualification (PQ) protocols.
- **Warning:** The cost of validating a self-hosted tool can exceed the cost of an Arena subscription for small-to-mid-sized companies.[31]

6. Conclusion

The migration from PTC Arena to an open-source PLM is a viable strategic move for organizations capable of managing their own infrastructure. **DocDokuPLM** is the recommended target due to its unencumbered AGPL license, modern cloud-native architecture, and strong alignment with the document-centric data model of Arena.

The feasibility relies on the **PDX standard** for extracting high-fidelity structure and the **REST API** for bulk file retrieval. However, the organization must be prepared for a fundamental shift in operational responsibility. The "convenience fee" of Arena is replaced by the "effort tax" of open source—requiring internal management of backups, security, and supply chain data integrations.

Summary of Recommendations

1. **Select DocDokuPLM:** It is the only modern, truly open option that fits the requirements.
2. **Use PDX:** Do not rely on CSV for BOMs; the structural loss is too high.
3. **Budget for Ops:** Allocate budget not for licenses, but for a DevOps resource to maintain the Dockerized stack.
4. **Validate Rigorously:** Use checksums and visual BOM comparisons to ensure the "digital twin" in the new system matches the legacy reality.

Works cited

1. BOMControl Solution Brief | Arena, accessed on November 28, 2025, https://www.arenasolutions.com/solution-brief/bomcontrol/
2. Arena Solutions PartsList and PDXViewer - Lifecycle Insights, accessed on November 28, 2025, https://www.lifecycleinsights.com/arena-solutions-pdxviewer/

3. Galvanize Harnesses Cloud PLM and AI for Enhanced Productivity - PTC, accessed on November 28, 2025, https://www.ptc.com/en/case-studies/galvanize-harnesses-cloud-plm-and-ai
4. PLM Data Migration - Razorleaf Corporation, accessed on November 28, 2025, https://www.razorleaf.com/services/plm-data-migration/
5. A Complete Guide to the 2025 Aras Community Edition, accessed on November 28, 2025, https://community.aras.com/blog/labs/a-complete-guide-to-the-2025-aras-community-edition/15785
6. New community edition (Release 2025) | Aras, accessed on November 28, 2025, https://community.aras.com/discussions/development/new-community-edition-release-2025/3458
7. New community edition (Release 2025) - Development - Aras Innovator, accessed on November 28, 2025, https://www.aras.com/community/f/development/55529/new-community-edition-release-2025/11279
8. Odoo Enterprise vs Community | Odoo Editions Comparison, accessed on November 28, 2025, https://www.odoo.com/page/editions
9. docdoku/docdoku-plm: The project purpose is to develop a comprehensive, robust open source PLM (Product LifeCycle Management) solution. - GitHub, accessed on November 28, 2025, https://github.com/docdoku/docdoku-plm
10. DocDokuPLM, accessed on November 28, 2025, https://www.docdokuplm.com/
11. Product Data Exchange (PDX) Definition - Arena Solutions, accessed on November 28, 2025, https://www.arenasolutions.com/resources/glossary/product-data-exchange/
12. How to export all attachments from a Fusion Manage tenant? - Autodesk, accessed on November 28, 2025, https://www.autodesk.com/support/technical/article/caas/sfdcarticles/sfdcarticles/How-to-export-all-attachments-from-a-Fusion-Manage-tenant.html
13. Transform Your Product Development With Arena's Powerful REST API Integration, accessed on November 28, 2025, https://www.arenasolutions.com/blog/introducing-arena-api-application-programming-interface/
14. Exporting to File - PTC Support Portal, accessed on November 28, 2025, https://support.ptc.com/help/servicemaxcore/en/articles/migration_tool/exporting-to-file.html
15. Export a PDX package - PDXpert PLM Software, accessed on November 28, 2025, https://www.buyplm.com/help-guide/pdx-package-part-bom-file-export
16. Arena API Solution Brief, accessed on November 28, 2025, https://www.arenasolutions.com/solution-brief/arena-api/
17. Download Aras Innovator, accessed on November 28, 2025, https://aras.com/en/download
18. Aras Innovator Community Edition License Agreement, accessed on November 28, 2025, https://aras.com/wp-content/uploads/2024/06/aras-innovator-community-edition

-license-agreement.pdf
19. List of Community Edition Apps (Free) - Odoo, accessed on November 28, 2025, https://www.odoo.com/forum/help-1/list-of-community-edition-apps-free-189585
20. Community version vs Enterprise Version - Odoo, accessed on November 28, 2025, https://www.odoo.com/forum/help-1/community-version-vs-enterprise-version-287832
21. Differences between Odoo Community & Odoo Enterprise, accessed on November 28, 2025, https://www.odoo.com/forum/help-1/differences-between-odoo-community-odoo-enterprise-239312
22. Docker configuration files for creating DocDokuPLM images - GitHub, accessed on November 28, 2025, https://github.com/docdoku/docdoku-plm-docker
23. openPLM - open source PLM, accessed on November 28, 2025, https://www.openplm.org/
24. REST API Version 2: Attachments - PTC Support Portal, accessed on November 28, 2025, https://support.ptc.com/help/servicemaxcore/en/articles/service_board/rest-api-version-2-attachments.html
25. Aras Innovator 2024 Release Query Builder Guide, accessed on November 28, 2025, https://aras.com/wp-content/uploads/2024/05/Aras-Innovator-2024-Release-Query-Builder-Guide.pdf
26. DocdokuPLM API, accessed on November 28, 2025, https://www.docdokuplm.com/docdoku-plm/rest-api-reference/
27. PLM Migration: A Guide to Best Practices & Strategies - Centric Software, accessed on November 28, 2025, https://www.centricsoftware.com/blog/plm-migration/
28. PLM Data Migration Strategies, accessed on November 28, 2025, https://www.plmnordic.com/plm-data-migration-strategies/
29. Step-by-Step Guide to Migrate PLM Data - Arena Solutions, accessed on November 28, 2025, https://www.arenasolutions.com/blog/step-step-guide-migrating-plm-data/
30. DataExtract Product Solution Brief | Arena, accessed on November 28, 2025, https://www.arenasolutions.com/solution-brief/dataextract-product/
31. SUPPORT Q&A - AGILE migration to Innovator - Archive - Aras Community, accessed on November 28, 2025, https://www.aras.com/community/f/archive/1010/support-q-amp-a---agile-migration-to-innovator