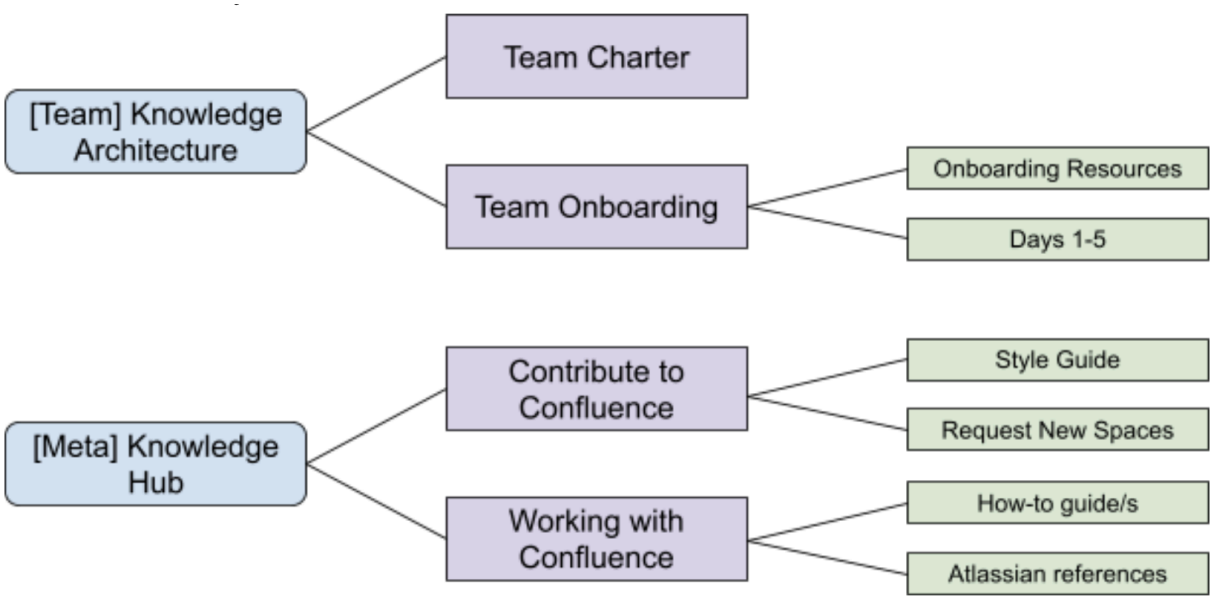
Confluence information architecture proposal

Proposal overview

Spaces, whether in Confluence or Notion, represent a grouping of alike items. It is tempting to think about spaces as hierarchical, matching the company's org chart or department structure, but the organization of people and teams is not 1:1 with the way information is created, curated, and ingested.

Instead, information can be considered stripped of team ownership as its most natural grouping, and considered in terms of subject matter or topic, as one might organize sections in a library.



Example: Team-specific vs. project-based spaces

Consider the division of two Knowledge Architecture spaces: a **[Team]-specific space** and a **team-agnostic/project space**.

If a user is looking for documentation on Product X (a specific product offering), they are not interested in the broader department as a whole, or docs related to team management and governance. Product X, as a product with its own set of requirements and documentation, could be a top-level organization item (space) for easily grouping related information.

Benefits of team-agnostic grouping

Outside of information retrieval, this team-agnostic grouping enables:

- **Better collaboration** in the case of cross-functional initiatives
- **Removes inherent assumptions of ownership** by separating content from team structure
- **Clearer information pathways** for users seeking specific topics

Confluence features such as page verification can be used to ensure a top-level ownership document is regularly up-to-date.

Knowledge Architecture guidance

The Knowledge Architecture team should provide guidance on what represents a good high-level Confluence space. Part of developing this recommendation could include:

- Auditing current products, projects, and existing workspaces
- Dividing them into theoretical spaces
- Providing guidance on when a document or subject belongs in a team space vs. a public space

---

Case studies

IT department

Initial expectation: One unified IT Space with directories for departmental teams; internal and external views of space information

Proposed structure:

Option 1:

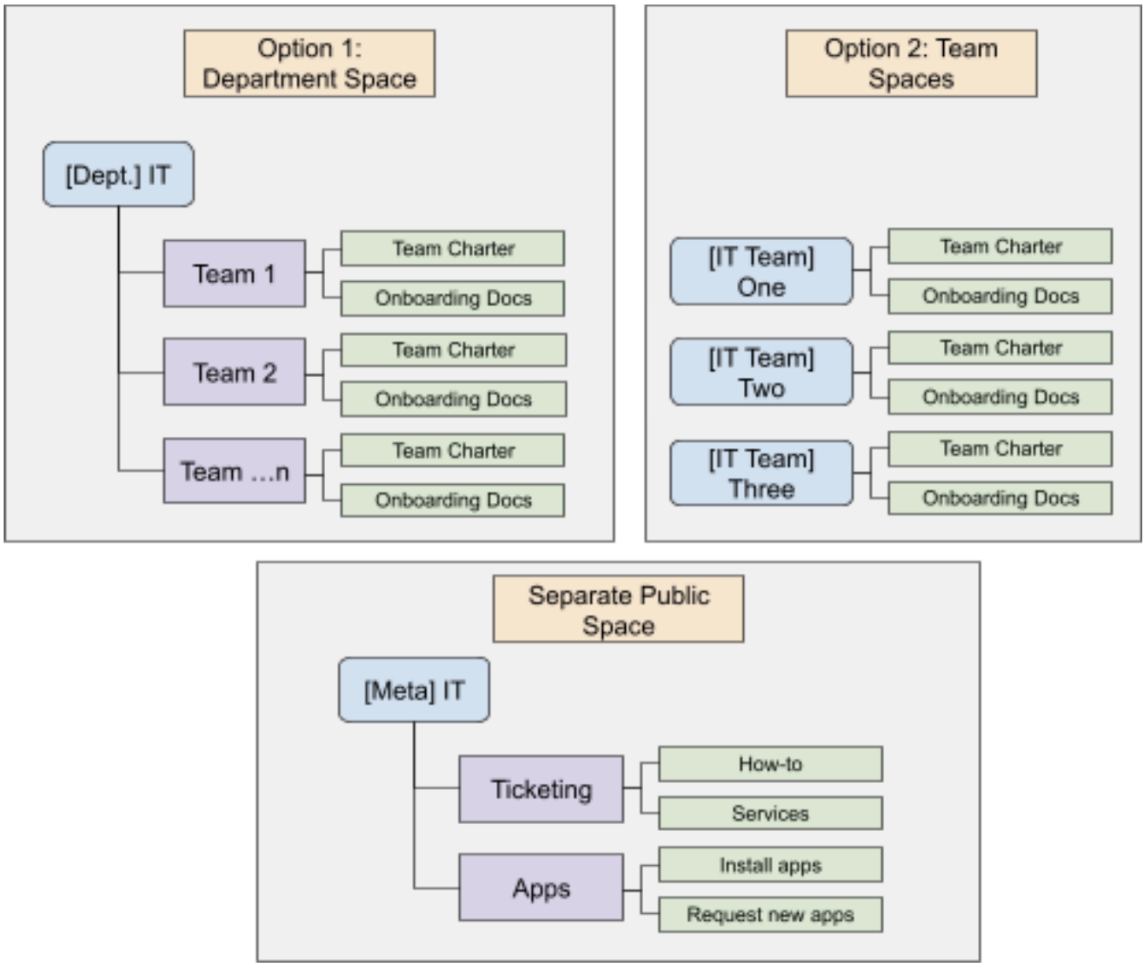
- IT department space with nested directories for individual teams
- Separate IT space for organization-wide resources
    - Department space should be prefixed or labeled to ensure it is not confused with external IT resources

Option 2:

- Individual IT team spaces with prefixes and/or labels (does not need to mirror org chart hierarchy)
- Separate IT space for organization-wide resources

Assumptions:

1. IT produces content which anyone in the organization should be able to access
2. Content might be managed or updated across IT or other teams
3. Content should not be confused with team-internal resources



Documentation team

Initial expectation: One unified Documentation team space with directories for various projects and documentation repositories

Proposed structure:

- Documentation Team space (for team processes, resources, and governance)
- Product Documentation space (for user-facing documentation projects)

Assumptions:

1. Teams outside of documentation may one day need to contribute to or understand workflows unique to documentation repositories
2. External contributors (e.g., Solutions Architects, Product teams) may create public-facing documentation

Product team

Initial expectation: Product team space encompasses both the product offering and the team structure

Proposed structure:

- Product Team space (for team management, processes, and internal resources)
- Product space (for product documentation, owned by all stakeholders)
    - Product space divided into product areas to better represent collaborative/multifunctional nature

Benefits: Separating team operations from product information allows cross-functional stakeholders to contribute without navigating team-specific content.

Recommendations

1. Clear and understood prefixing of spaces

Use consistent prefixes to distinguish space types:

Examples:

- `[Team] Product Team` vs. `[Product] Product Name`
- `[Team] IT - Team Name` vs. `IT Resources`
- `[Department] Engineering` vs. `[Project] Project Name`

2. Robust labeling

Implement a comprehensive labeling system to:

- Categorize content by topic, audience, or purpose
- Enable advanced filtering and search
- Support cross-functional discovery

3. Page metadata and ownership fields

Create pages with example metadata or fields for describing:

- Ownership (DRI - Directly Responsible Individual)
- Content type (process, guide, reference, etc.)
- Audience (public, team-specific, department-specific)
- Last review date

4. Page verification for ownership documents

Use page verification on any project or team ownership pages to ensure they are periodically updated by the DRI. This helps maintain accuracy and accountability.

5. Public space standards

Consider developing standards for governing the use of spaces:

- What makes a team space? (Team-internal processes, meeting notes, etc.)
- What makes a project space? (Cross-functional initiatives, product documentation)
- What makes a long-lasting public space? (Company-wide resources, policies, knowledge base)

Problems to solve/avoid

Information discoverability

Problem: "Where do I find X?"

Solution: Clear space structure, robust search, and consistent labeling to ensure users can easily locate information.

Reproducing existing information systems

Problem: Simply recreating the current information architecture from previous tools (e.g., Notion) without improvement.

Solution: Use the migration as an opportunity to redesign information architecture based on user needs and best practices.

Sensitive information management

Problem: Handling confidential or sensitive information (e.g., MNPI - Material Non-Public Information) appropriately.

Solution:

- Develop guidance around linking to secure storage (e.g., Google Drive) rather than uploading sensitive documents directly
- Clearly label spaces and pages containing sensitive information
- Determine how to prevent indexing of confidential pages in Confluence

Search indexing control

Problem: Ensuring certain pages are not indexed in search results.

Solution: Determine best practices for controlling page visibility and search indexing in Confluence.

Implementation questions

Administrative considerations

Questions to address:

- What tasks must be performed by administrators?
    - Which parts of these cannot be delegated?
    - Which parts can be documented for self-service?
    - Which parts can be broken down to provide the smallest amount of work possible for administrators to implement?

User expectations and training

Questions to address:

- For individuals who have never used Confluence before, what are their expectations?
    - How do they expect spaces to be organized?
    - What do they expect sidebars to contain?
    - How do they expect to navigate pages?
    - What are their expectations around searchability?
    - How do they expect nesting and hierarchy to work?

Action item: Provide examples and training materials to demonstrate how information is structured and accessed in Confluence.

Next steps

1. **Audit existing content:** Review current products, projects, and team spaces to identify natural groupings
2. **Develop space taxonomy:** Create a clear taxonomy for space types, prefixes, and labels
3. **Create governance guidelines:** Document when to create new spaces vs. adding content to existing spaces
4. **Design templates:** Build page templates with metadata fields for ownership and categorization
5. **User testing:** Validate the proposed structure with representative users from different departments
6. **Training and documentation:** Create resources to help users navigate the new structure
7. **Iterate:** Gather feedback post-implementation and adjust as needed

---

Conclusion

By organizing Confluence spaces around **subject matter and projects** rather than strictly mirroring organizational hierarchy, we can create a more intuitive, collaborative, and scalable information architecture. This approach:

- Improves information discoverability
- Supports cross-functional collaboration
- Reduces confusion around ownership
- Creates clearer pathways for users to find what they need

The key is providing clear guidance, consistent standards, and ongoing governance to ensure the structure remains useful as the organization grows.