Case Studies
Application Ownership Framework

This case study presents two scenarios demonstrating how the Application Portfolio Framework supports organizations in addressing business challenges through structured catalogs, defined relationships, and practical properties. The framework enables clarity in application ownership, hosting, and lifecycle management.

Framework Summary: Application Portfolio & Ownership Model

This framework provides a brief structured approach to managing an organization's **application portfolio**, including ownership and hosting details. It captures the essential properties of applications, their technical custodians, and their hosting locations, while also mapping the relationships between these elements.

Core Components

Application

- Maintains the inventory of applications within the enterprise.
- Key properties include:
    - **Criticality** – Importance of the application to the organization.
    - **Recommendation** – Lifecycle recommendation for the next three years (e.g., maintain, replace, upgrade).
    - **Cost** – Indicator of application cost on a scale of 1–5.
    - **End of Life** – Date when vendor support for the application ends.
    - **Business Fit** – Alignment with business objectives: 1-5
    - **Technical Fit** – Alignment with enterprise technology standards and policies: 1-5

Actor

- Captures the list of **technology custodians** or technical owners responsible for applications.
- Properties include **email** and **phone** contact details.
- Provides accountability and a clear ownership model.

Location

- Defines the **physical site** where applications are hosted, typically datacenter names.
- Supports transparency of infrastructure deployment and hosting strategy.

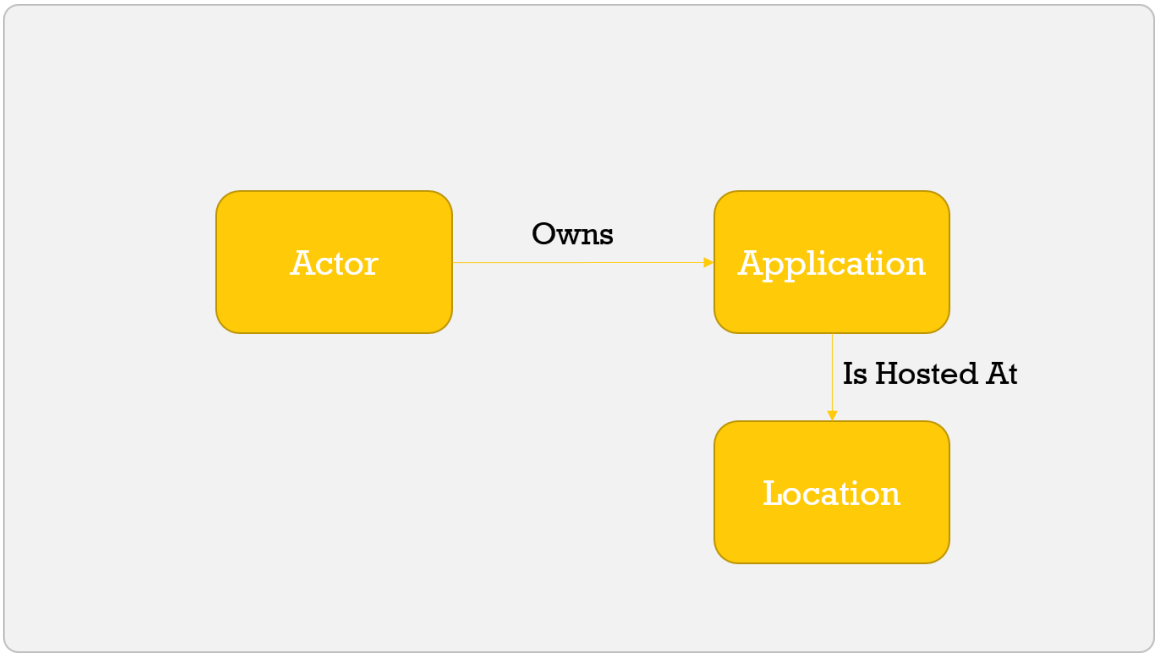
Relationships

Actor Owns Application

- Defines the relationship between a technical custodian (Actor) and the applications they own.
- Ensures responsibility and traceability in application management.

Application Is Hosted At Location

- Defines where each application is physically deployed.
- Useful for operational visibility, compliance, and disaster recovery planning.



Scenario 1: Sarah in the Banking Sector

Problem Faced

Sarah, an Enterprise Architect in a large banking organization, faced challenges with overlapping applications performing similar functions across departments. The lack of

visibility into application criticality, business fit, and end-of-life timelines led to increased costs and operational risks.

Framework Application

The framework's **Application Catalog** provided a structured view of applications, capturing critical fields such as Criticality, Recommendation, Cost, Business Fit, and Technical Fit. The **Actor Owns Application** relationship established clear ownership by linking applications to custodians, while the **Application Is Hosted At Location** relationship revealed hosting dependencies.

Deliverables Produced

- Application portfolio catalog with lifecycle recommendations
- Ownership matrix linking applications to responsible actors
- Hosting map showing dependencies across datacenters

Business Value Created

In addition to the proper awareness about the Applications' baseline, the bank reduced redundant applications, optimized licensing costs, and mitigated risks by proactively planning application retirements. The structured framework enabled decision-makers to align IT investments with business priorities, enhancing operational resilience.

Scenario 2: Tomson in the Manufacturing Industry

Problem Faced

Tomson, an IT Director in a global manufacturing company, struggled with fragmented application hosting across multiple sites. Each division has its own list, and applications were deployed inconsistently, making it difficult to assess costs, align with technology standards, or ensure business continuity.

Framework Application

A consistent **Application Catalog** allowed Tomson to have a unified view on the portfolio and assess Business Fit, Technical Fit, and Costs across applications. By leveraging the **Application Is Hosted At Location** relationship, his team identified hosting inefficiencies and dependency risks. The **Actor Owns Application** relationship clarified accountability, enabling smoother IT operations.

Deliverables Produced

- Application-hosting inventory with datacenter mapping
- Cost analysis across the application landscape
- Responsibility assignment through the actor-application relationship

Business Value Created

In addition to the proper awareness about the Applications' baseline, the manufacturing company consolidated hosting locations, reducing infrastructure costs and improving system availability. Clear accountability streamlined support processes, while the improved application insights ensured compliance with technology policies and long-term alignment with manufacturing business goals.