Avyka

Case Study

Migrating to Harness IDP 2.0: From Inline Chaos to a Governed, Remote-First Catalog

CUSTOMER OVERVIEW

The client is a data-driven enterprise with a mature internal developer portal (IDP) practice, managing a growing catalog of software components and APIs through Harness IDP. As their engineering footprint expanded, the need to modernize their catalog management approach, moving from inline to remote configuration, became a critical initiative to ensure version control, governance, and long-term maintainability.

When Harness released IDP 2.0, the client saw an opportunity to align their catalog infrastructure with the latest standards. They needed a trusted implementation partner to navigate the migration with precision and minimal disruption.

Understanding the Challenges

The platform team was operating on an inline catalog configuration that lacked version control and scalability. Transitioning to IDP 2.0 meant navigating a newly released framework with its own set of early-stage limitations, while keeping critical pipelines operational.

Inline Configuration Debt

All catalog entities (Components and APIs) were stored inline, with no remote version control in place.

Syntax Overhaul Required

The existing YAML structure was built on older Backstage conventions and required a full rewrite to conform to IDP 2.0 specifications.

Pipeline Fragility

The primary "All In One" pipeline, responsible for IDP OpenAPI spec validation and API definition generation, began failing immediately after IDP 2.0 was enabled.

Framework Gaps

IDP 2.0, being newly launched, did not yet support multi-branch entity storage within a single repository or custom user group creation at the time of implementation.

Script Dependency

The client's existing Python scripts for catalog entity lifecycle management (create, update, delete) were incompatible with the new IDP 2.0 framework.

What began as a structured migration quickly became a live troubleshooting exercise. With most IDP functionalities failing post-enablement and the primary validation pipeline non-operational, the client needed immediate, expert-led resolution, not just implementation support.

The Avyka Approach

Avyka took full ownership of the migration, partnering directly with the Harness IDP expert team to resolve framework-level limitations while simultaneously delivering on all client requirements.

Solutions

- **IDP 2.0 Enablement:** Raised a Harness support ticket to activate IDP 2.0 within the client's account, initiating the automatic conversion of existing entities, scorecards, and plugins.
- **Manual Catalog Migration:** Transitioned all catalog entities, Components and APIs, from inline to remote configuration for robust version control.
- **YAML Re-architecture:** Rewrote entity YAML structures, syntax, and spec blocks to fully comply with IDP 2.0 standards, moving away from legacy Backstage conventions.
- **Python Script Modification:** Overhauled the client's existing Python scripts to support entity lifecycle management within the new IDP 2.0 framework.
- **Pipeline Validation:** Triggered the "All In One" pipeline via a GitHub PR, successfully passing all IDP OpenAPI spec and API definition checks post-migration.

What We Did

As Harness released updates supporting multi-branch entity storage and custom user groups, these capabilities were immediately integrated to close remaining use-cases.

Re-architected the Catalog

Migrated all Component and API entities from inline to remote, version-controlled configuration.

Resolved Framework Gaps

Collaborated with the Harness IDP team to identify workarounds and integrate new features as they became available.

Delivered End-to-End Validation

Ensured the primary "All In One" pipeline and all PR checks passed successfully upon migration completion.

Results

The migration delivered a stable, future-ready IDP catalog environment with every client objective met:

- **Full Catalog Migration:** All Component and API entities successfully transitioned from inline to remote, version-controlled configuration.
- **Pipeline Restored & Validated:** The "All In One" pipeline, including IDP OpenAPI spec and API definition checks, passed successfully following migration.
- **Zero Blocked Use-Cases:** All requirements, including multi-branch entity storage and custom user group creation, were fulfilled upon feature availability.
- **Increased Client Trust:** Avyka's ownership of the Python script modifications and proactive issue resolution significantly strengthened the client relationship.
- **IDP 2.0 Ready:** The client now operates on a fully compliant, scalable IDP 2.0 environment positioned for future growth.

ABOUT AVYKA Avyka is a dedicated system integrator specializing in harnessing the full potential of Harness to revolutionize cloud infrastructure and enhance DevSecOps practices.