

Shilpa Murthy

Lead UX Design & Research Advocate



Projects

Developer Sandbox

3- Site Replication – IBM Flash Systems

UX Research – IBM Storage Defender



Case Study 1

Developer Sandbox



Developer Sandbox

The Developer Playground project [reimagines how developers discover, test, and evaluate APIs by creating a fully interactive, sandboxed environment embedded directly within the API Hub ecosystem](#).

Rather than forcing developers to download SDKs, configure local environments, or struggle through fragmented documentation, the Playground offers an immediate, guided, and hands-on experience—allowing users to run sample apps, execute real API calls, explore code, customize solutions, and even commit changes to their own Git repositories.

My role in this project was end-to-end and deeply strategic. [I led the definition of the vision and user experience for the Playground, shaped the interaction flows, and crafted the conceptual model that brings APIs, tutorials, live execution, and code exploration into a single unified workspace](#). Based on research that revealed developers overwhelmingly prefer sandbox environments to evaluate APIs, I designed the core user journeys—from API discovery to experimentation to purchase—and ensured that each step reduced friction and increased confidence for developers.

I created the UX architecture for how APIs appear inside the Playground, including:

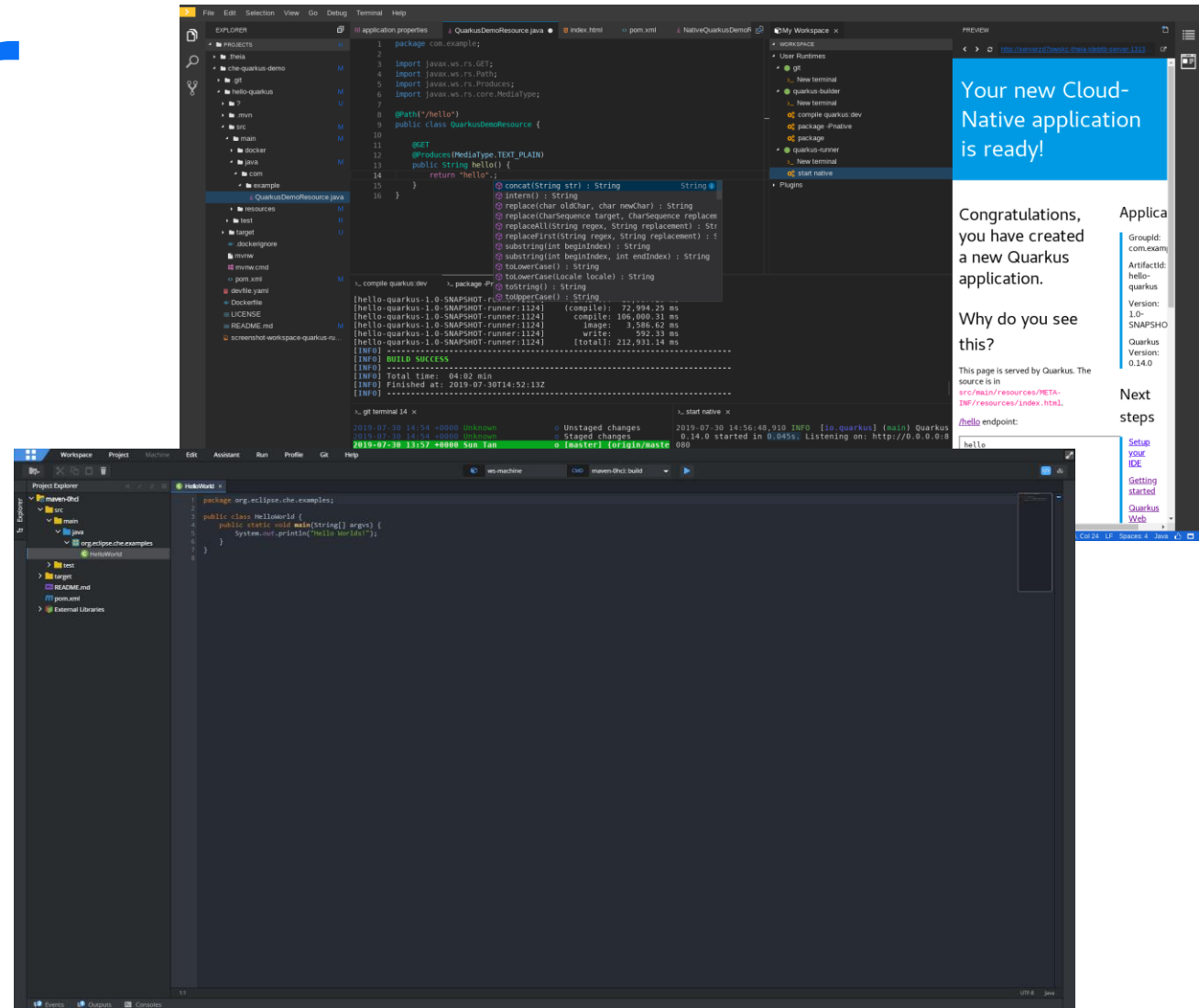
- The left-panel resource exploration model (Selected API, Recommendations, History)
- The didact/tutorial integration that auto-generates step-by-step instructions
- The live execution panel that builds and runs sample apps instantly

Developer Sandbox

Platform

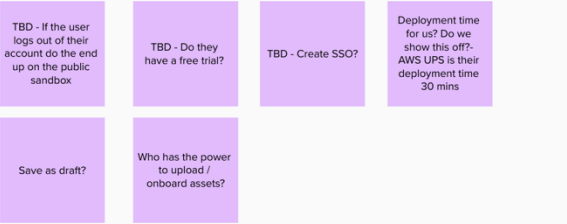
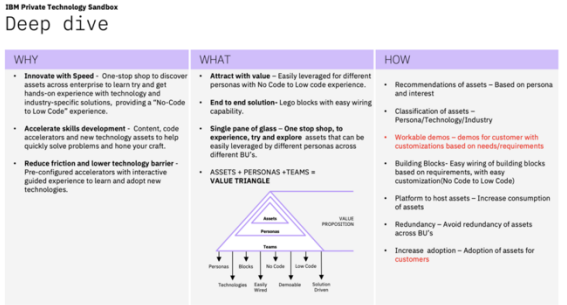
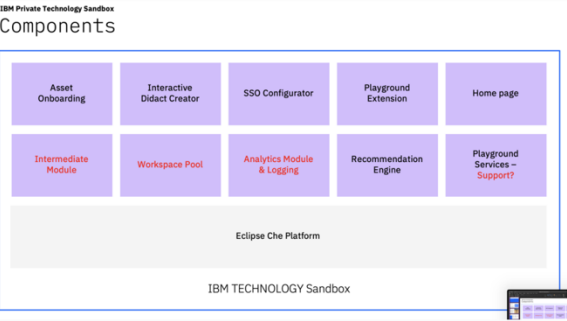
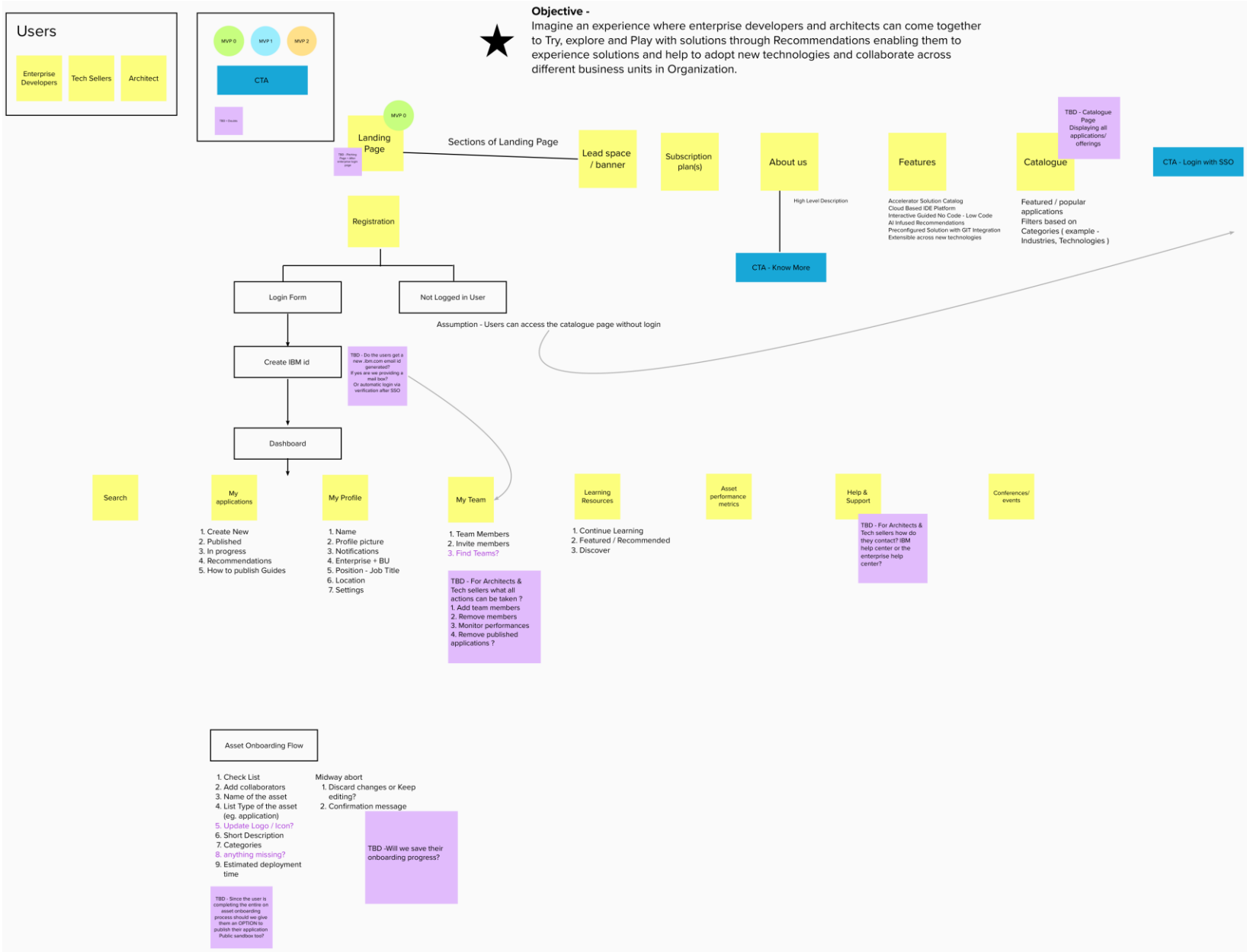
Eclipse Che is a cloud-based, browser-accessible integrated development environment (IDE) that provides full workspaces, dev environments, and tools without requiring any local setup.

It allows developers to write, build, and run applications entirely in the cloud using containerized workspaces.



User Journeys Workflows

Brainstorming



Brainstorming

Basic Workflow

Home page

Catalogue

Details Page

Intermediate Page

Workspace IDE

Labels

MVP 0

MVP 1

MVP 2

Objective

This information with the detailed user roles & tasks is needed to identify the user's work flow for the private sandbox and visually represent this information in the UI and build a meaningful experience.

These pages will be owned by IBM Developer

Pages	Assumptions & Questions	User Roles & Tasks	Comments	Resources/ Ref materials
Home Page	Assumptions Building this as a template, we are identifying the common placeholders for the page. Currently all these are in IBM branding & we are listing the following. For any first time user these are the sections that they will see - 1. About us 2. Features 3. Applications 4. Tutorials 5. Help & Support Questions 1. What all sections are incorporated in the private sandbox homepage ? 2. For any first time user - how do they land on this page? 3. If they haven't logged in how are the different roles identified on this page ? 4. For private sandbox what applications are we surfacing here? 5. How is the content driven to this page ?	Joe - Bigtech Admin Bob - Acme User Please list the user roles & tasks for the above mentioned user Please list the user roles & tasks for the above mentioned user V- Assumption: Here is that there is a white label space for this entity to have this information shown. Currently IBM Developer has a few partner spaces, but I'm not sure what information lives there. What types of applications would be built and would they be IBM applications or the business partners that leverage IBM products and there's? If this doesn't live on Developer, where is this housed, how do partners get access? White label spaces have a tendency to be behind authentication - what is the logged in experience going to be? Should an anonymous user even know this exists?	Please add any comments or remarks or questions along with your name in this column	
Catalogue Page	Questions 1. What applications are surfaced on the catalogue page for the first time users ? 2. How do we define categories of the filters Example Technology filters Industry filters Domain filters	Add inputs for the questions listed MVP 0		
Details Page	This section would be identified based on the above two sections Questions 1. How is the content driven for this page ?		G1: Do we need to provide this. This link the content can go straight to the intermediate page. If they Private Sandbox adapter wants to provide this go already have it in their own CMS they can.	
Intermediate Page	Assumptions For private sandbox we need to just give placeholders for the content and one CTA to launch the IDE	V- Assuming this page is the page that is shown while the sandbox is loading - what content would need to change to make this relevant to the white label space? Should it be customized? How can it be customized?	G1: For this I think we just need to be able to extend the IBM part but keep it generic beyond that for MVP	
IDE	Assumptions For the first time private sandbox users they will see the customized IDE with IBM branding (The one that is currently available today) Questions 1. What all can the users customize in the IDE? 2. Are we building a template for all of the following Didact Builder Sample Applications Left Navigation Bar in Eclipse che Header & Footer	V- Assuming that the menu has been integrated and the branding has been attached to the showhide function of the menu item - the menu items would also need to be taken into consideration and make sure that the ones that are used in the public sandbox are relevant to the white label space.	G1: What do you mean by IBM Branding? If you mean things that say "IBM" they should be configurable https://github.com/ibm-developer-private-sandbox/ibm-developer-private-sandbox G2: Long single Didact builder should be available. MVP we need to have a more manual way for them to provide them to where they are stored by Joe G3: Sample apps need come from github.com from an organization that the "Provider" center. Joe manages V: How is access provided from private apps for the editors to review and edit? Where do assets/labels stored? How can the repo be linked to keep the content current should copyrights be made?	

WHY HOW WHO WHAT

Some big questions haven't been answered - and should be, before moving forward .

What is the purpose of the private sandbox?

How is it different from IBM Garage?

Where does this fit into the flow of supporting the 5 or 6 sales pillars?

What are the metrics that can be used to gauge the success of this private sandbox?

How is the content going to be sourced?

Who is going to edit and review the content?

How is this going to be integrated into IBM Developer? If not, where will this live?

How will a user be able to access this? Through invite?

How is this going to be tied to the IBM Developer profile for tracking progress?

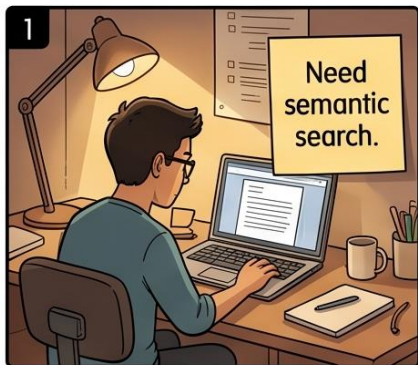
Please add any comments or remarks or questions along with your name in this column

IBM Developer publication flow should be taken into consideration for both private and public sandbox environments - and the dependencies for producing content.

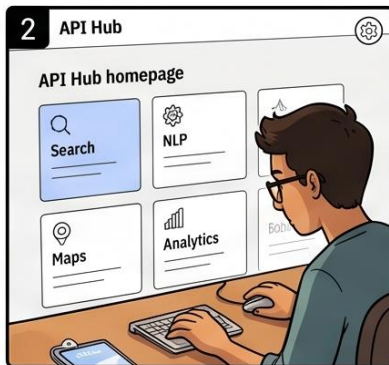
A-IS: Initial content requester submits a content intake form > form creates an issue in Git for editors to review and accept or reject. > If accepted, Editor will use their own text editor and Git to track edits and make changes to the content as it was submitted. Magic happens and content is published on IBM Developer.

To edit one of these applications and the associated content - Editors will need access to the repo that houses the application and all the other assets required for publication. What are those artifacts, what do the artifacts need to contain, and where will they be stored for publication?

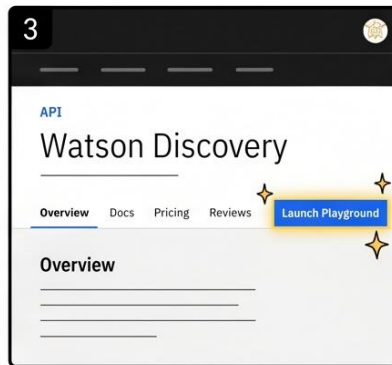
Vision



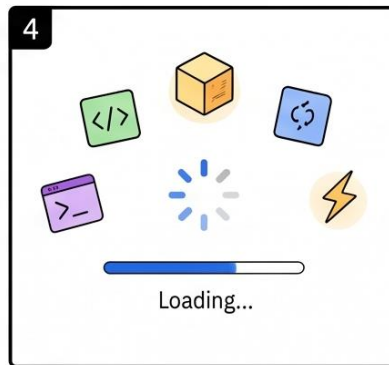
Alex is looking for an API that solves semantic search for their product.



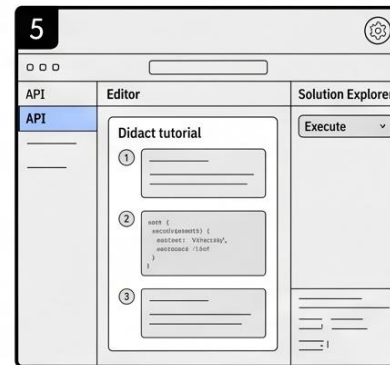
Alex browses the API catalog to explore available solutions.



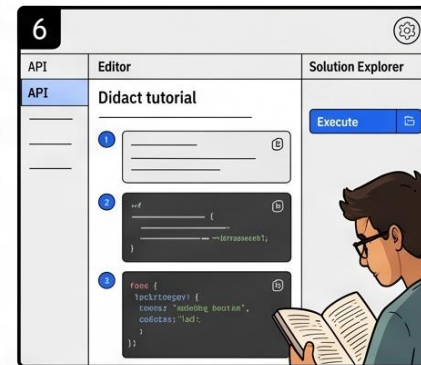
Watson Discovery catches Alex's attention.



The Playground prepares a ready-to-use sandbox environment.



The Playground opens with the Watson Discovery API loaded.



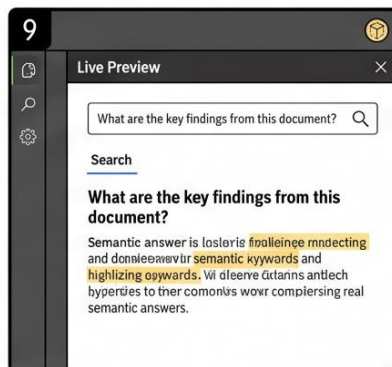
The tutorial guides Alex step-by-step.



A ready-to-run sample app is provided.



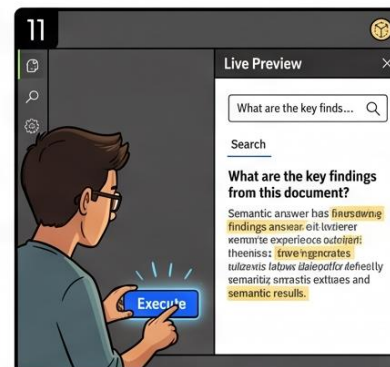
With one click, Alex runs the sample app in the sandbox.



The app runs instantly, showing real semantic search results.



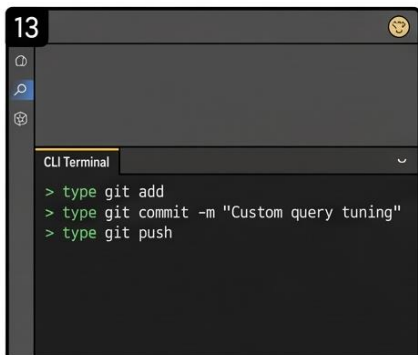
Alex modifies the query parameters to fit their domain.



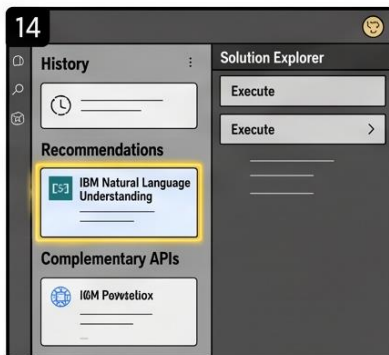
Alex quickly tests and validates changes.



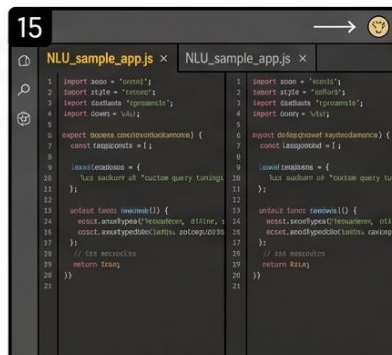
Alex links their GitHub account to save their custom app.



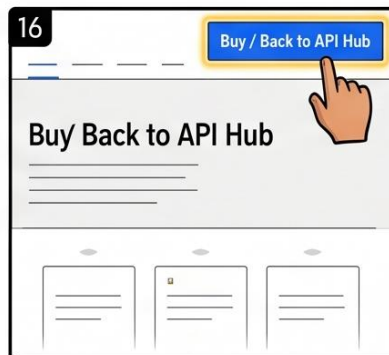
Customizations are saved to Alex's personal repo.



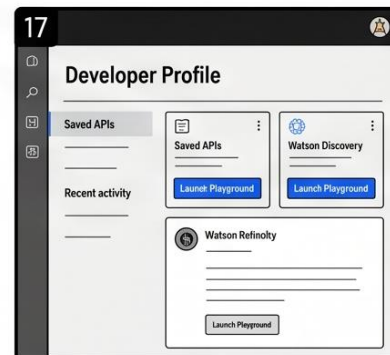
The Playground recommends complementary APIs.



Alex evaluates another API to compare features.



Confident in the fit, Alex returns to purchase the API.



Later, Alex returns to continue prototyping.



In one session, Alex went from discovery to a working prototype.

Outcome Statement

Designed and delivered a zero-setup, interactive Developer Sandbox that enables developers to instantly discover, execute, and experiment with APIs in a unified environment—reducing time-to-first-call, increasing hands-on engagement, and accelerating conversion from exploration to adoption.

Wireframes Designs



Prototype & Design Explorations

<https://drive.google.com/file/d/1VPyuiDjNIVcQ2DuzIBwxuWCl-54aDzBM/view?usp=sharing>



Case Study 2

3 Site Replication IBM Flash
Systems



3 Site Replication Flash System

A storage administrator configures a [3-site replication setup for a financial system that cannot tolerate downtime](#).

The system guides the administrator through defining a topology where [two metro sites operate in synchronous high availability, while a third remote site maintains an asynchronous disaster recovery copy](#).

During a network partition between the HA sites, the system automatically ensures data integrity by designating a single active site and clearly communicating the state to the administrator.

Once connectivity is restored, the administrator is guided through a seamless recovery workflow to re-establish full replication.

This experience transforms a traditionally complex and error-prone process into a predictable, transparent, and resilient system.

1 – High-stakes Environment (healthy state)



A storage administrator is responsible for a payment processing system that cannot go offline and must be protected against local failures and regional disasters.

2 – Problem Trigger



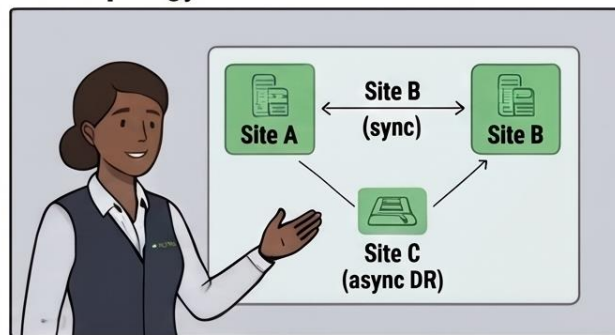
The administrator realizes that a 2-site setup is not sufficient to handle regional outages. Introduce motivation for 3-site setup.

3 – Entry into Workflow



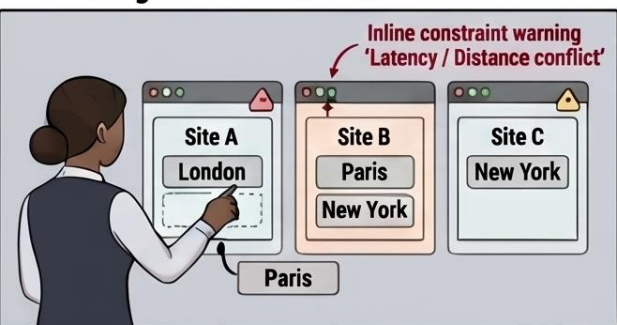
Instead of configuring replication manually, the system asks: "What outcome do you want?" The administrator selects 3-site HA + DR.

4 – Topology Visualization



Visualizing the selected topology: Site A & B: Active Active availability; Site C: Disaster Recovery (Async DR).

5 – Configuration in Context



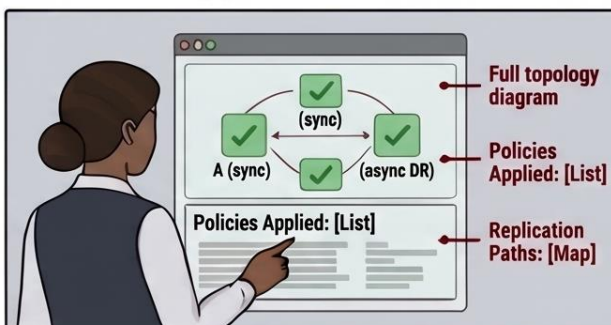
As the administrator configures locations, the system validates for low latency. Inline constraint warning of 'Latency/Distance conflict' is shown with a red icon.

6 – Workload / Partition Mapping

Volume	Workload	Protection
Payment Database		→ 3-site HA
Reporting Data		→ DR only

Assignment of volumes/workloads to protection levels. A critical Payment Database receives 3-site HA protection. Map workloads: Payment DB → 3-site protection/data → DR only.

7 – Confirmation State



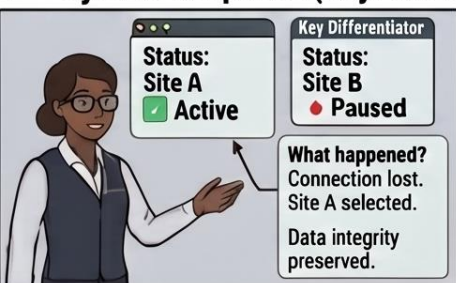
A comprehensive summary is shown before deployment. The admin confirms.

8 – Failure Event



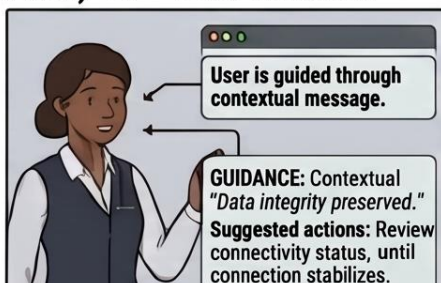
Network failure breaks sync between HA sites.

9 – System Response (Key Differentiator)



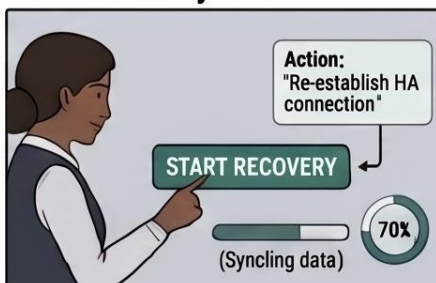
Explain What Happened: Connection lost between Site A & B. **Decision:** Site B selected; data integrity preserved.

10 – User Guidance



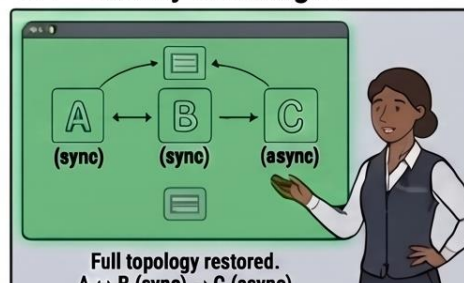
Guided resolution: Context message & actions, reduced stress.

11 – Recovery Flow



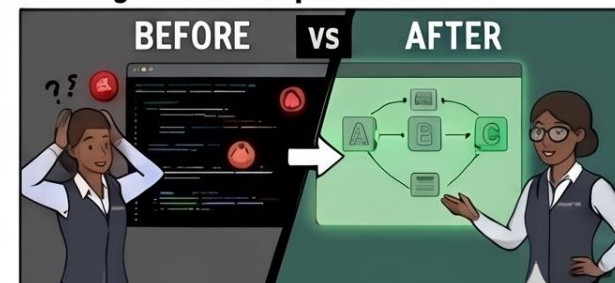
Guided re-sync: progress visual, reduced stress & structured process.

12 – Steady State Again



Returns to full 3-site resilience with no manual complexity.

Closing Frame – Impact Statement



Before: A high-risk, expert-only process → fragmented and complex CLI workflow.
After: Into a guided, resilient, and explainable workflow.

Outcome Statement

A clear, guided replication management experience that enables storage administrators to confidently configure, monitor, and recover 3-site replication setups with assured data integrity and minimal operational effort.



Case Study 3

Research Storage Defender



Research
Storage Defender

Transform DRS into a unified, intelligence-driven recovery platform that delivers clarity, confidence, and coordinated action across cyber recovery operations.

Unify the Recovery Ecosystem

- Bring recovery planning, grouping, dependencies, threat analysis, scanning, and clean rooms into a single, coherent experience.
- Eliminate fragmentation across backup products, storage systems, and manual processes.
- Provide one authoritative view of applications, recovery points, and system health.

Clarify and Guide

- Resolve terminology confusion across applications, recovery groups, phases, controlled environments, and threat states.
- Add contextual guidance, micro-instructions, onboarding moments, and step-level prompts to reduce cognitive load.
- Present meaningful visualizations with clear hierarchy, readable labels, accessible colour choices, and intuitive navigation.

Elevate Intelligence and Automation

- Deliver actionable threat insights: what was detected, where, its type, affected hosts, and expected next steps.
- Automate dependency ingestion, scanning, validation, testing, and documentation to reduce manual effort.
- Support exceptions with manual overrides, custom recovery ordering, placeholder applications, and progress tracking.

