PRODUCT CASE STUDY · PLATFORM ARCHITECTURE · HAPPY PATH STUDIOS

HPS Product Framework

Shared architecture, applied with judgment about what should stay specific.

Role: Product Strategy · UX · Platform Architecture · Development

Ownership: Happy Path Studios

The HPS Product Framework is the reusable application foundation behind focused products built at Happy Path Studios. It exists to reduce repeated product infrastructure work while keeping each product specific to its own audience and operational need. Plantation Paw Prints is the concrete example used throughout this case study.

The problem was not building one system that every product had to fit. It was deciding what deserved to be shared, and what needed to stay specific to the product in front of me.

01 / THE PROBLEM

New focused products keep needing the same foundation.

Every new focused product tends to need the same underlying capabilities: a way to manage content, handle media, authenticate users, manage permissions, take payments, administer the product, and deploy it. None of that is specific to any one product.

Rebuilding that foundation from nothing for every new product adds real cost. It also means each product's foundation evolves on its own, disconnected from what the last product already learned.

The question

How do you remove the cost of rebuilding common product infrastructure for every new product, without treating every product as if it were the same product?

The HPS Product Framework exists to remove that repeated cost, without treating every product as if it were the same product.

02 / THE ARCHITECTURAL PRINCIPLE

Reuse where it creates leverage. Stay specific where it does not.

The framework is not built around maximum abstraction. Its guiding decision is where to draw the line between shared architecture and product specific behavior.

Reuse

Architecture is shared across products only where that reuse creates meaningful leverage: proven patterns that already work, applied again instead of rebuilt from nothing.

Product Specificity

Every product still keeps its own behavior and experience specific to its own audience and operational need, wherever generalizing that experience would add complexity the product does not actually require.

The resolution

Every product could be built entirely from nothing, or forced entirely into one identical shape. Neither extreme serves the product or the person using it. Successful reuse requires judgment about both what should be shared and what should remain specific to the product.

03 / SHARED FOUNDATIONS

Reusable capability areas, not a fixed product shape.

The framework provides reusable patterns across a defined set of capability areas. A new product draws only on the capability areas it actually needs. No product is expected to use every area, and the framework does not assume otherwise.

- Content Management
- Authentication
- Commerce
- Administration
- Fulfillment
- Deployment
- Media
- Permissions
- Payments
- Orders
- Operational Workflows

Individual products use the capabilities appropriate to their own needs. Plantation Paw Prints, the applied example later in this case study, draws on several of these areas and not all of them.

04 / FROM FRAMEWORK TO PRODUCT

Proven foundations, applied to a specific audience.

A new product starts from capability areas that already exist and already work, rather than from nothing. What makes it a focused product rather than a copy of the last one is everything built specific to its own audience and its own operational need: the experience, the workflows, and the decisions that only make sense for that product.

Content & Media

Content Management, Media

Structured content and media handling reused across products.

Identity & Access

Authentication, Permissions

Shared patterns for who can sign in and what they can do.

Commerce & Fulfillment

Commerce, Payments, Orders, Fulfillment

Reusable patterns for payment, orders, and fulfillment.

Operations & Deployment

Administration, Workflows, Deployment

Shared patterns for running and deploying a product.

Product Layer

Product Specific Experience

Each product decides which shared capability areas to use, then builds the rest, the actual experience, workflows, and decisions, specific to its own audience and operational need.

Applied Example

Plantation Paw Prints

A club platform built on the shared foundation: content, media, commerce, and administration among the capability areas it draws on. One applied example, not the whole framework.

05 / PLANTATION PAW PRINTS

One focused product, built on the shared foundation.

Plantation Paw Prints began as a greenfield project and became a working club platform: public content, CMS capabilities, events, a calendar, community features, media management, commerce, Stripe checkout, order tracking, and production and fulfillment workflows, supported by role based administration.

Plantation Paw Prints is one concrete example of the framework applied to a focused product. It does not exercise every capability area the framework provides, and it is not meant to.

The story that matters is not how many features Plantation Paw Prints contains. It is how reusable foundations supported a product that still feels specific to its own audience and purpose.

06 / PRODUCT EVIDENCE

What one focused product built on the framework became.

Plantation Paw Prints is one concrete example of the framework applied to a focused product. These screens are representative, not a complete inventory of the product.

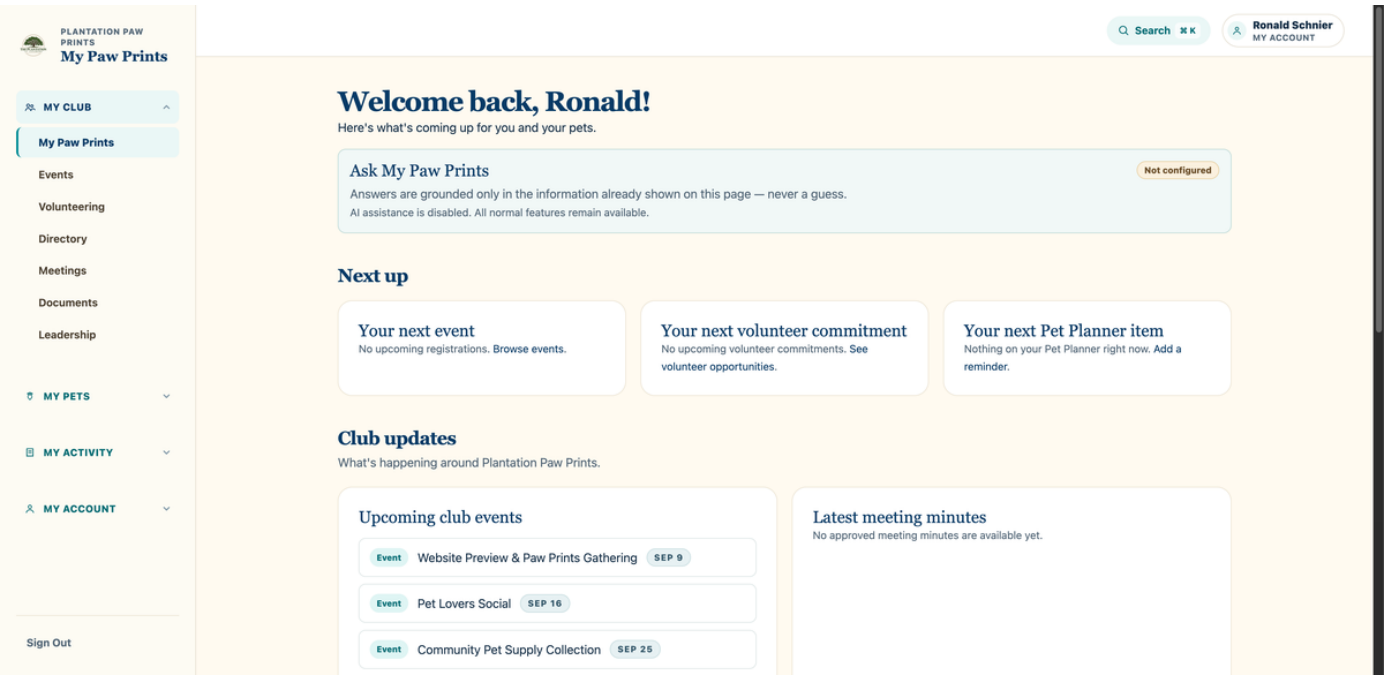
Public Experience

The public facing club experience: content, events, bulletin information, member stories, merchandise, causes, volunteering, galleries, and participation, all built from the shared foundations of the framework. Shown here as a preview of the top of the homepage rather than the complete page.

06 / PRODUCT EVIDENCE

Member Experience

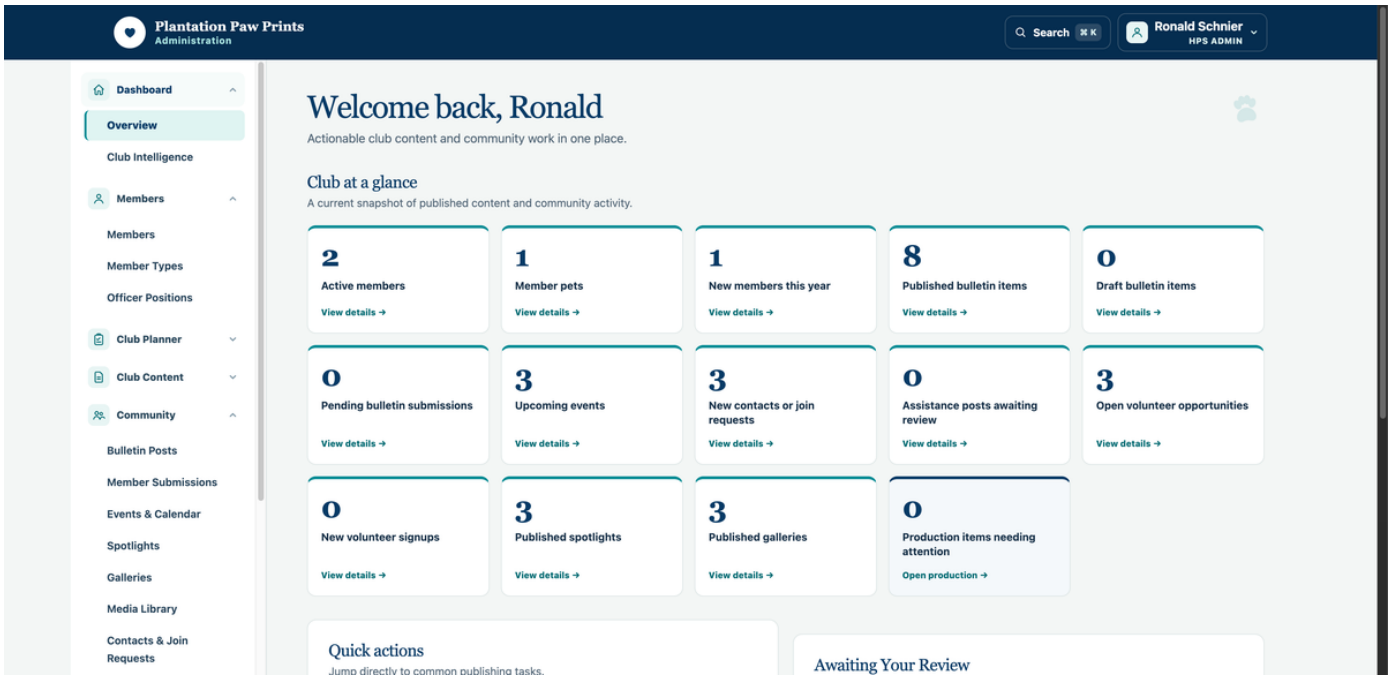
The authenticated member experience, showing that the framework extends beyond the public website into a signed in product surface.



06 / PRODUCT EVIDENCE

Administration

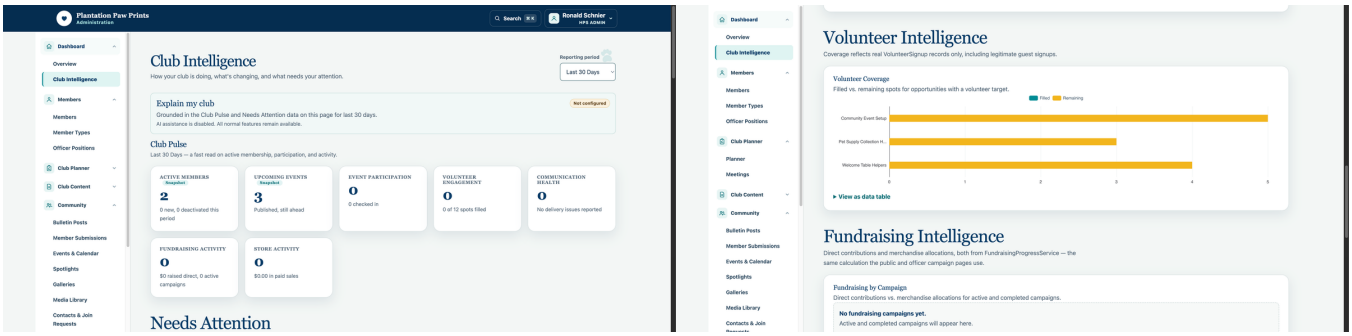
The operational layer behind the public site and member experience: managing members, pets, content, events, community activity, media, contacts, and production related work.



06 / PRODUCT EVIDENCE

Operational Intelligence

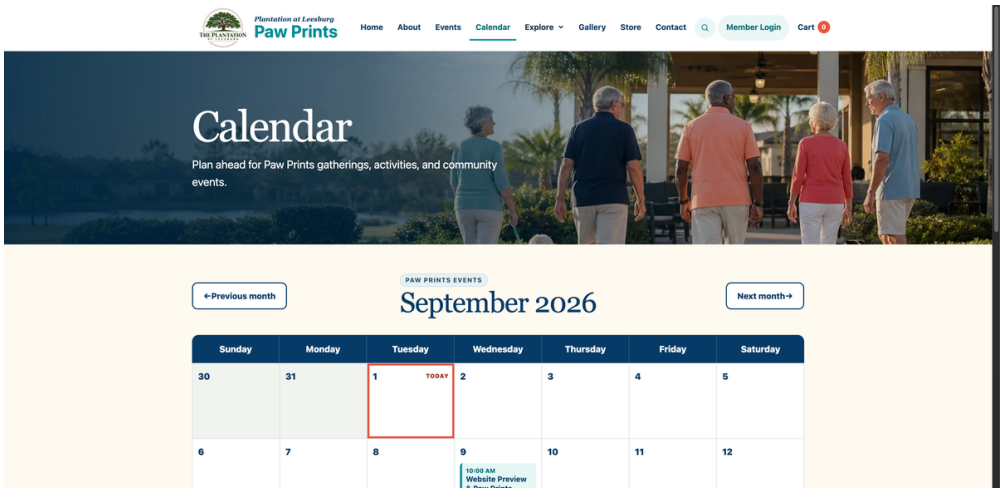
Beyond administering content, the product also supports operational understanding and decision support. A summary layer and a deeper layer beneath it are part of the same operational intelligence rather than two separate features.



06 / PRODUCT EVIDENCE

Focused Feature Example

A focused feature built on the same shared architecture: a calendar experience purpose built for this product rather than a generic component applied everywhere.



07 / THE PRODUCT DECISION

Plantation Paw Prints stayed pilot first.

The product decision

Plantation Paw Prints was not generalized into a multi tenant SaaS product before that need was proven.

That was an intentional product and architecture decision, not a limitation. Multi tenant architecture is not the wrong choice in general. It is the wrong choice before the need for it has actually been demonstrated. Staying pilot first kept the product focused on what it needed to prove first.

Abstraction that has not earned its place adds cost without adding value. The judgment here was not about what the architecture could support. It was about what this specific product actually needed at this stage.

08 / WHAT I OWNED

Product strategy through platform architecture and development.

- Product Strategy
- UX
- Platform Architecture
- Development

09 / OUTCOME AND LEARNING

The foundation, and the judgment about what not to generalize.

The framework provides a reusable foundation for creating focused products while preserving the ability to make product specific decisions for each one.

The broader learning is that successful reuse depends as much on knowing what not to generalize as it does on identifying what can be reused.

What this case demonstrates

Reusable platform architecture, product judgment about what should be shared and what should stay specific, an applied product proof through Plantation Paw Prints, and ownership spanning product strategy, UX, platform architecture, and development.

Ronald Schnier

Senior UX / Product Designer

Portfolio: ronaldschnier.com

Prepared as a focused product design case study supplement.