



privacyforms.studio

**Enterprise-grade web forms with total data sovereignty,
modern SurveyJS integration, and optional AI acceleration.**

Andreas Jung
Plone Conference 2026 · Maastricht
www.zopyx.com - www.andreas-jung.com

SPEAKER

Andreas Jung

Plone & Python consultant · software architect · form-wrangler since the pre-Volto era

PLONE ARCHAEOLOGIST

Two decades around Plone, Zope and Python.

*Still knows where the old
form bodies are buried.*

INTEGRATION PLUMBER

FastAPI, databases, APIs, PDFs, exports, auth.

*If data leaks, blocks, or deadlocks,
it becomes my problem.*

AI CO-PILOTED BUILDER

Architecture, code and design discussed with AI — then reviewed by humans...and sometimes the Meat Proxy.

AI slop, but with tests and taste.



Personal mission: make forms boringly reliable — and prevent Germany from reinventing the fax as a web app.



Andreas Jung / ZOPYX is an official SurveyJS Partner.

Designed, built, and challenged with AI

Privacy Forms Studio is not only a forms project — it is also a case study in AI-assisted software engineering.

IMPLEMENTATION

Designed and implemented with AI assistance.

ARCHITECTURE

Discussed, iterated, and challenged with AI.

VERIFICATION

Reviewed critically — not accepted blindly.

THIS IS AI SLOP — at its best.

🌟 LIVE DEMO

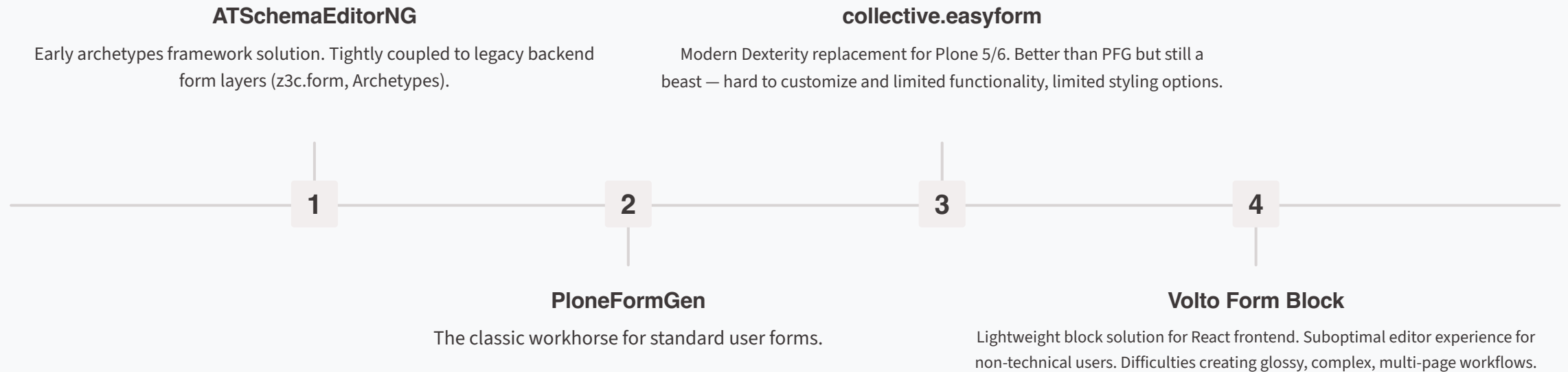
Try it live! ✨

See Privacy Forms Studio in action — no setup, use demo login, no friction.

demo.privacyforms.studio



Evolution of forms in Plone



COMMON ARCHITECTURAL CHALLENGES

Legacy Coupling

Tight dependencies on deprecated backend form layers make migration and modernization costly.

Limited Customization

Restricted layout and styling options hinder enterprise-grade UI requirements.

Editor Experience

Suboptimal interfaces for non-technical users and difficulties building complex multi-page workflows.

Motivation for Privacy Forms Studio

Digital Sovereignty

EU-driven compliance mandating full control over sensitive data without third-party SaaS dependencies.

Complex Requirements

Increasing client demand for beautiful, multi-step digital forms with complex conditional logic and validation.

AI Capabilities

Leveraging modern AI to accelerate form creation while maintaining strict on-premise trust boundaries.

Top Design Objectives

User & Admin Focus

- Modern replacement for EasyForm and legacy builders.
- Visual drag-and-drop authoring workspace.
- Complex branching, logic, and multi-page flows.
- Multilingual and accessible (WCAG compliant) by default.

Architecture Focus

- Self-hosted deployment ensuring full data sovereignty.
- Native Plone integration with native security & roles.
- Open, portable JSON-based form definitions.
- Support for different security models.
- Seamless integration via REST APIs and Webhooks.
- Security and professionalism by design.

Feature Highlights

Visual Designer

Intuitive drag-and-drop form builder for rapid development.

Conditional Logic

Powerful skip rules and expression evaluation for complex workflows.

Privacy-First

Zero external tracking or required SaaS calls — full data sovereignty.

PDF Digitization

Convert paper processes to streamlined digital workflows.

Enterprise Governance

Designed specifically for strict GDPR environments and enterprise compliance requirements.

Multilingual Support

Available in ten languages with full RTL support.

Optional AI Support

AI-accelerated form generation and assistance — fully optional, humans stay in control.

Scalability

Supports small and large-scale Plone installations for low- and high-volume form and poll traffic.



Let's see it in action!

A quick video demonstration of Privacy Forms Studio.



Privacy Forms Studios

layers protections across

rendering and submission.

Powered by SurveyJS Engine

SurveyJS provides an industry-standard, client-side rendering engine and visual creator component.

By combining SurveyJS with Plone, Privacy Forms Studio delivers enterprise governance without reinventing form UX.

30+ Input Types

Support for dynamic matrices, dropdowns, files, and ratings.

JSON Schema

Clean, portable declarative JSON schema for forms and responses.

Calculated Fields

Dynamic expressions, value variables, and inline validation.

Responsive UI

Mobile-optimized and WCAG accessible out of the box.

32 Predefined Themes

Light & dark variants plus panelless layouts — 32 theme variations out of the box.

Design Token System

5-layer CSS variable architecture: palette → base → primitives → semantic → component tokens.

Visual Theme Editor

No-code Theme Editor in Survey Creator generates reusable JSON theme objects for brand alignment.

Shared Across Products

One theme JSON applies consistently to Form Library, Creator, Dashboard, and PDF Generator.

CORE TECHNOLOGY

SurveyJS — What is it?

SurveyJS is a set of client-side JavaScript libraries for building a full-cycle form management system on your own infrastructure. Every form is defined by a portable JSON schema — no vendor lock-in, no third-party data storage.

CORE

FREE

Form Library

Free & open-source (MIT). Renders dynamic, JSON-based forms in React, Angular, Vue, or plain JS — with conditional logic, validation, and branching. — **the heart of Privacy Forms Studio.**

CORE

COMMERCIAL

Survey Creator

Visual drag-and-drop builder that generates portable JSON schemas. Fully white-label and customizable. Commercial license. — **used for visual form authoring in Plone.**

OPTIONAL

Dashboard

Visualises survey results with interactive charts and tables. Interprets JSON schemas to populate custom dashboards. Commercial license.

Optional — not currently used in Privacy Forms Studio.

OPTIONAL

PDF Generator

Renders any SurveyJS form as a PDF, with editable or pre-filled export. Commercial license.

Optional — PDF export handled differently in Privacy Forms Studio.

❏ Form Library is free (MIT). Survey Creator, Dashboard, and PDF Generator require a commercial license — one-time per developer. Supports React, Angular, Vue, jQuery, and Vanilla JS. Fully WCAG / Section 508 / ARIA compliant. No usage limits — unlimited forms, responses, and users.

SurveyJS Licensing Model

CORE

Form Library Viewer

Production / Runtime

€0 Free

Unlimited forms, submissions, and users.

CORE

Survey Creator Basic

Development / Authoring

€499 one-time per developer

Optional renewals.
Visual drag-and-drop builder.

OPTIONAL

SurveyJS PRO Suite

Enterprise / Advanced

€899 one-time per developer

Includes PDF generator and dashboard and full suite.



Zero licensing costs for end-users, content editors, or form respondents.

Editors using the "Creator" component (editor) do not need a dedicated SurveyJS license.

Integrators using Privacy Forms Studio count as integrators/developers and **require a license.**

One integrator/developer with multiple rollouts = one license per integrator needed.

Financially attractive for larger organizations.

Made in Estonia/EU.

Technical Architecture

A clean separation of concerns across every layer of the stack.

Layer	Technology Choice	Key Functionality
Frontend	Plone 6 Classic & SurveyJS	Seamless form rendering, validation, and visual design workspace.
Definition Format	Open JSON Schema	Portable definitions decouples forms from underlying database framework.
Storage Engines	ZODB or RDBMS (SQLAlchemy)	Support for local Plone ZODB or dedicated external relational SQL databases.
Delivery & API	REST API & Webhooks	Real-time submission routing, email notifications, and background processing.

The background features a dark gray grid with a central vertical line. Overlaid on this are several wavy, translucent lines in shades of teal, purple, and blue, which appear to be composed of many fine, overlapping strokes. The overall effect is a sense of depth and movement.

Deep dive...

The Foundation Pipeline

One JSON definition flows through authoring, rendering, and response collection.

1

1. Form JSON

Single declarative JSON definition containing fields, validation, logic, and localized labels.

2

2. Survey Creator

Visual drag-and-drop authoring environment for form designers and editors.

3

3. Form Library

Free runtime engine rendering interactive forms and collecting responses in client browsers.

Survey access modes controls the entrance

Server-side policy: who may access and submit a form or poll?



PUBLIC

Open URL

Polls · contact · registration

Controls:

- 🧑‍🔧 Rate limits
- Validation
- Submission tokens



TOKEN LINK

Bearer URL

Possession grants access

Controls:

- Cryptographic token
- Expiry
- Scope



TOKEN CONTAINER

Managed token object

Lifecycle + audit trail

States:

- valid
- used
- revoked / expired



Every request is checked by the backend. Standard Plone roles, groups, permissions and workflows can be applied where needed.

Strong validation on the backend in order to avoid abuse or malicious data being submitted.

Abuse Protection

Layered controls applied before a submission is accepted.

Short-Lived Tokens

JWT access tokens expire quickly, reducing the useful lifetime of a leaked token or to avoid replay attacks.

Dual Validation

The same form rules are checked in the browser and again on the server.

One-Time Submission

A token is consumed after use; repeated submissions with the same token are rejected.

Mass-Submission Defence (planned)

Token gating prevents simple replay. Rate limiting remains a deployment issue.

Implemented: token expiry · dual validation · single-use consumption | Planned / deployment-specific: rate limiting

Submission Validation Pipeline

Every submission passes a 10-step server-side hardening pipeline before it is stored, mailed, or forwarded.

1. Transport Limits

Oversized payloads rejected with HTTP 413 before JSON parsing.

2. JSON Shape

Submission must be a valid JSON object; primitives and arrays rejected.

3. Schema Fields

Only fields from the active form schema accepted; unknowns rejected.

4. Text Safety

Blocks XSS, dangerous markup, script injection and on* event handlers.

5. Data URLs

Only Base64 PNG/JPEG allowed; malformed or unsupported URLs rejected.

6. File Structure

Files must have valid filename, MIME type and content members.

7. Filenames

Unicode NFC normalized; path traversal and control characters rejected.

8. MIME & Content

Allowlist enforced; magic bytes verified for all supported formats.

9. Resource Limits

File count and decoded size limits enforced independently.

10. Canonical Hand-off

Normalized deep copy passed to validator and downstream only.

 **Force Server Side Validation (default: on)** — compiled SurveyJS binary (automatically compiled using Deno), Submission contents and secrets are never logged.

Governance

Policy, traceability, and control over form definitions and submissions

Version Control

Form Versions Are Versioned

Every form definition is fully versioned — changes are tracked, previous versions can be restored, and audit trails are maintained for compliance.

- Full change history per form
- Restore previous definitions
- Compliance-ready audit trails

Audit Trail

Complete Submission Logging

All submissions and form changes are logged with timestamps and user attribution for full traceability.

- Timestamped submission records
- User attribution on every action
- Tamper-evident log storage (planned)

Access Control

PLONE

Role-Based Permissions

Leverages Plone's native security model — roles, groups, and permissions govern who can create, edit, publish, or view forms.

- Plone roles & groups
- Per-form permission policies
- Fine-grained view/edit control

Workflow Integration

PLONE

Content Workflow Participation

Forms participate in Plone content workflows — publish, retract, or archive forms as part of standard editorial processes.

- Standard Plone workflows
- Publish / retract / archive states
- Editorial process integration

Actions

Actions define what happens after a submission is accepted — any combination of `store`, `mail`, `mail-notification`, and `post` can run simultaneously.

`store` — Persist the submission

Saves the submission to the results store (**ZODB** or **RDBMS**) with metadata including `poll_id`, `created`, `user`, and `answer` payload. ZODB enabled by default.

`mail` — E-mail exported results

Sends export files (PDF by default) **as e-mail attachments**; 10 different formats available and configurable.

`mail-notification` — Notification e-mail

Sends a **lightweight notification** e-mail with a link to the submission detail view — no attachments, no data.

`post` — Forward to HTTP endpoint

POSTs the validated payload as a webhook to a configured HTTP(S) endpoint.

EXPERIMENTAL

AI Assistance

Optional AI acceleration that keeps humans in the loop and data on-premise.

All AI features are fully optional — zero AI is the default. You choose if, when, and where AI is used.



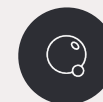
Prompt → Form

Draft complete forms from a simple text description.
Describe structure, fields, and logic in plain language — AI generates the JSON schema instantly.



Document → Form

Convert existing documents into digital forms: upload mockups, scanned PDF forms, or plain-text specifications — AI extracts fields, types, and structure automatically.



Chatbot Assistant

An integrated chatbot helps users navigate Privacy Forms Studio: answers questions, guides configuration, and explains features in plain language.

Infrastructure & Governance

Local AI Support

Integrates with Ollama for zero data leakage.
Any local or remote AI provider supported.

Human Control

AI generates drafts; human editors review & publish.

- 📄 Full control: use no AI, a local model (Ollama), or any remote provider — configured per deployment. You decide the level of AI involvement. Most decent models work well, including lightweight flash/small models — no need for the largest or most expensive ones.

Data Handling & Processing

Enterprise-grade data management with flexible storage, multi-format export, and automated hand-off capabilities.

Flexible Storage

Store submissions in **ZODB** or **relational databases** via SQLAlchemy, or route directly **without persistent storage**.

Multi-Format Export

Export response data instantly to **10 different formats** like PDF, XLSX, CSV, DOCX, HTML, Markdown, XML, and JSON formats.

Automated Hand-off

Trigger secure **webhooks**, custom Python processing pipelines, and configurable multi-recipient emails. Support of **Plone notifications/events**.

Deployment & Storage Strategy

Keep definitions in Plone. Configure submission and caching backends independently for each Plone site.

✓ Plone Core

- Form definitions
- Version history
- Permissions & workflows

Plone 6.2 remains the authoritative configuration layer.

✓ Small Deployments

- Standalone or ZEO
- Submission data in Plone
- Shared SQLite cache

A simple setup is sufficient for low traffic and modest form usage.

⚠ High-Volume Use

- RDBMS submissions
- Shared RDBMS-backed cache
- Avoid ZODB write contention

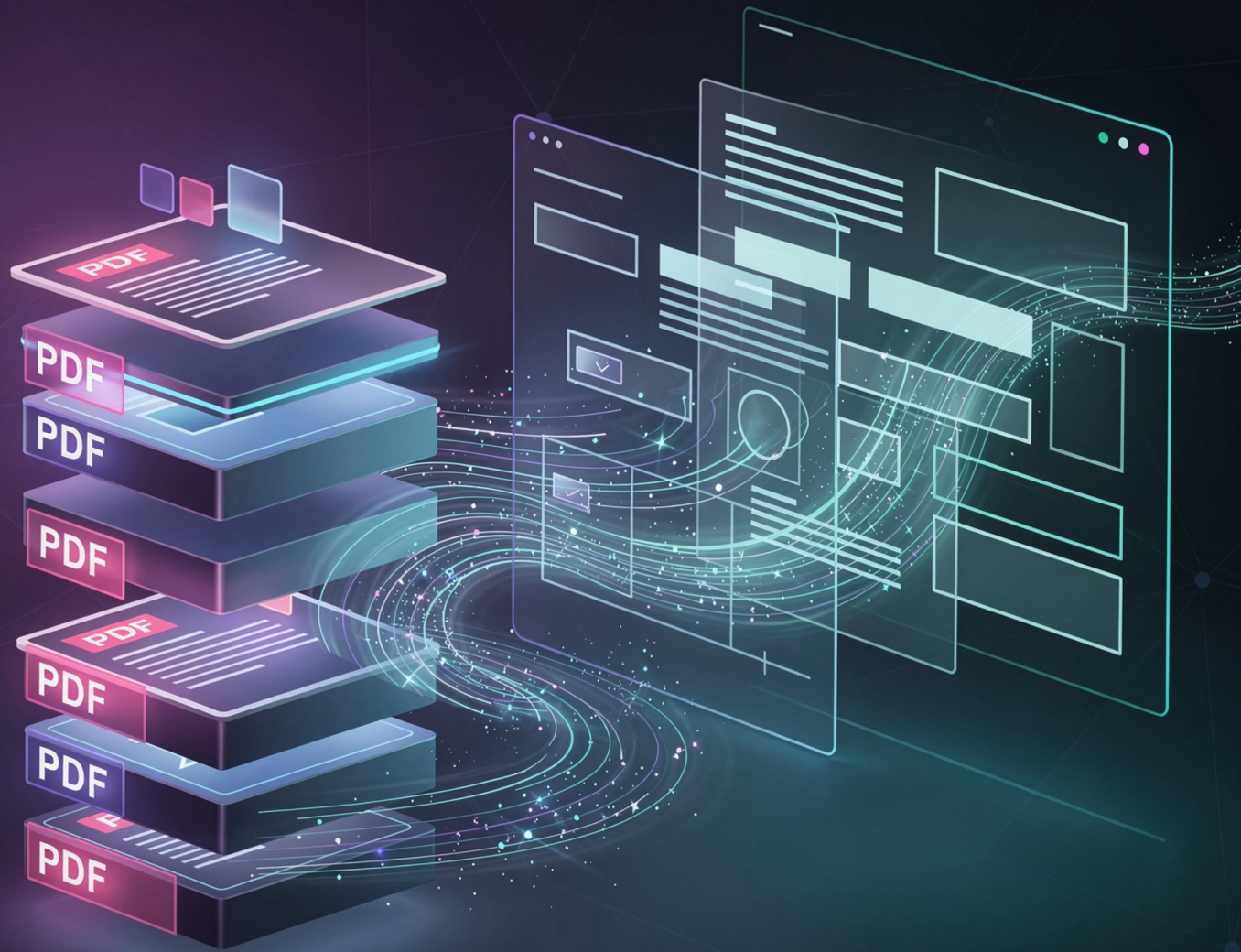
Designed for busy forms, polls and multi-node deployments.



Per-site configuration: Submission and caching backends are configured independently for each Plone site.



Recommendation: Use an RDBMS when traffic or concurrency grows.



Digital workflows and PDF forms

Use Case: PDF to Web Form

Transform legacy paper-based processes into modern, accessible digital workflows.



Ingest Paper Forms

Scan legacy paper or PDF forms into the system.



Smart Extraction

AI extracts form fields, titles, and section structures.



Interactive Web Form

Generates accessible, multi-page SurveyJS web forms.



Structured Data

Responses captured as clean JSON for backend processing.

Workflows with fillable PDFs 1/3

"Fillable PDF forms are a plague — limited functionality, poor UX, and still printed out anyway."



Limited Functionality

Fillable PDFs can't branch, validate properly, or adapt to user input.



Poor User Experience

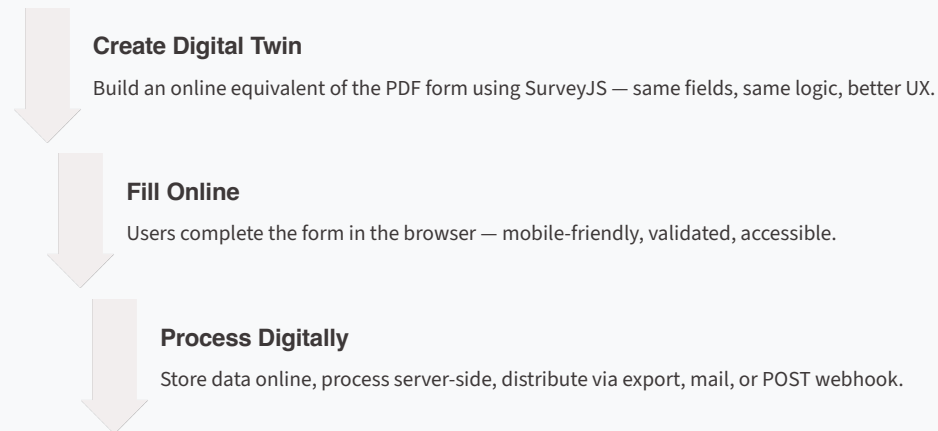
Clunky, inconsistent across viewers, and not mobile-friendly.



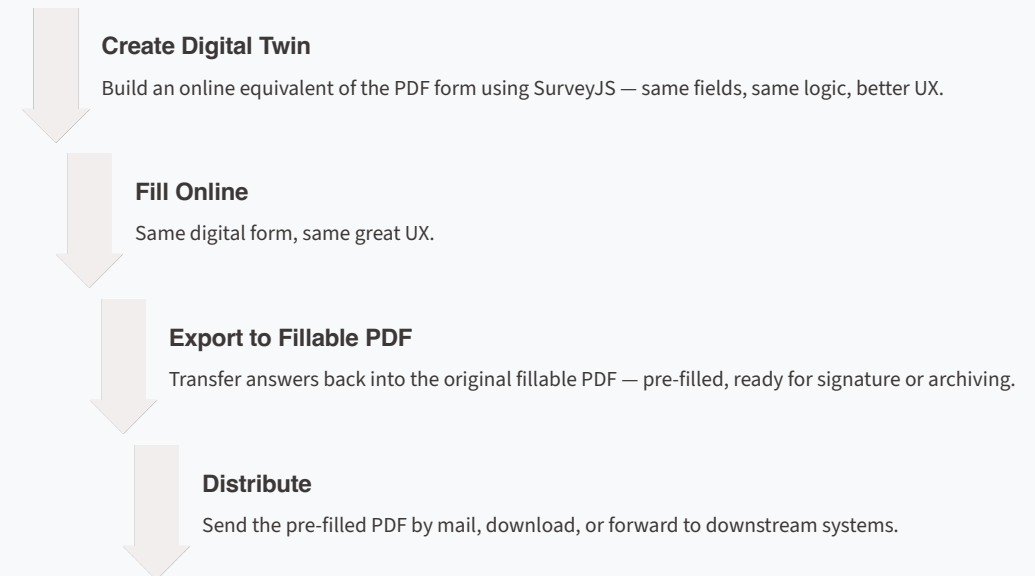
Still Gets Printed

Despite being "digital", most fillable PDFs end up printed and filed anyway.

Digital Twin Workflow



PDF Round-trip



 **The best of both worlds:** modern online UX with PDF compatibility for legacy workflows.

Workflows with fillable PDFs 2/3

What we extract from the PDF

Form Fields

Name, type, value, and options.

Geometry

Position, size, and page number.

Field Groups

Visual row clusters preserving layout logic.

Non-field Content

Labels, headings, and annotations with context.

How the Digital Twin is built

01

Extract

pypdf parses AcroForm fields, geometry, and RowGroups.

02

Map to SurveyJS

Field types are mapped and names sanitized.

03

Enrich with Context

Labels are linked to fields via spatial proximity.

04

Generate Online Form (LLM-supported)

SurveyJS JSON schema produced with pdfFieldName for round-trip mapping.



Tools: pypdf · privacyforms_pdf (AcroForm parsing, geometry, RowGroups) · pdfcpu (deterministic field extraction, reference pipeline)

Workflows with fillable PDFs 3/3

Filling a PDF with submitted form data is a two-step process: validate & extract with pypdf, then write values back with PyMuPDF.

pypdf · Validate & Extract

Upload Validation

Checks the PDF has AcroForm fields via `get_fields()`. Rejects non-form PDFs early.

Field Extraction

Reads field names, types (`/FT`), flags (readonly/required), options (`/Opt`) and widget rects from `/Annots`.

Mapping Check

Each PDF field is checked against the SurveyJS form — marking which fields have a digital twin counterpart.

PyMuPDF · Fill & Return

Receive Form Data

Submitted field values arrive from the browser (text, checkbox, combobox, listbox).

Write to Widgets

PyMuPDF sets `widget.field_value` per field name and calls `update()` for each widget type.

Return Filled PDF

A pre-filled `<name>_filled.pdf` is returned — ready for download, signature, or archiving.



Field matching is by name — the SurveyJS form and the PDF must share identical field names for the fill to work correctly.

Open Issues

Which license?

Which license can this AI slop have in which legislation (EU, USA, world-wide)?

Experimental functionality

PDF fillable form workflows (digital twin creation, PDF fill-back via PyMuPDF) and AI-related functionality are experimental and subject to change.

AI Assistance

Prompt-to-form, document-to-form and chatbot features require further refinement before production use.

Volto support

Unlikely and of little interest — Privacy Forms Studio targets Plone Classic; a Volto integration is not planned.

SurveyJS Theming

SurveyJS 3 introduced a complete new theming approach and implementation is currently being adjusted

collective-ai-settings

Looking into new AI settings.
PFS based on own [privacyforms.ai](#) module.

CONCLUSION

Key Takeaways



Professional & Decent Solution

For building form-driven applications and services.



Affordable & Open

Almost free to deploy — open-source core with transparent, affordable commercial licensing for advanced features.



Digital & Data Sovereignty

Ensuring your complete control over your data.



(optional) AI Assistance Under Your Control

Leverage powerful AI tools with full oversight and management.



Full Control Over Data & Processing

Maintain command throughout the entire data lifecycle.



LIVE DEMO

Try It Yourself

Explore real forms built with Privacy Forms Studio. The demo resets every 6 hours.

7

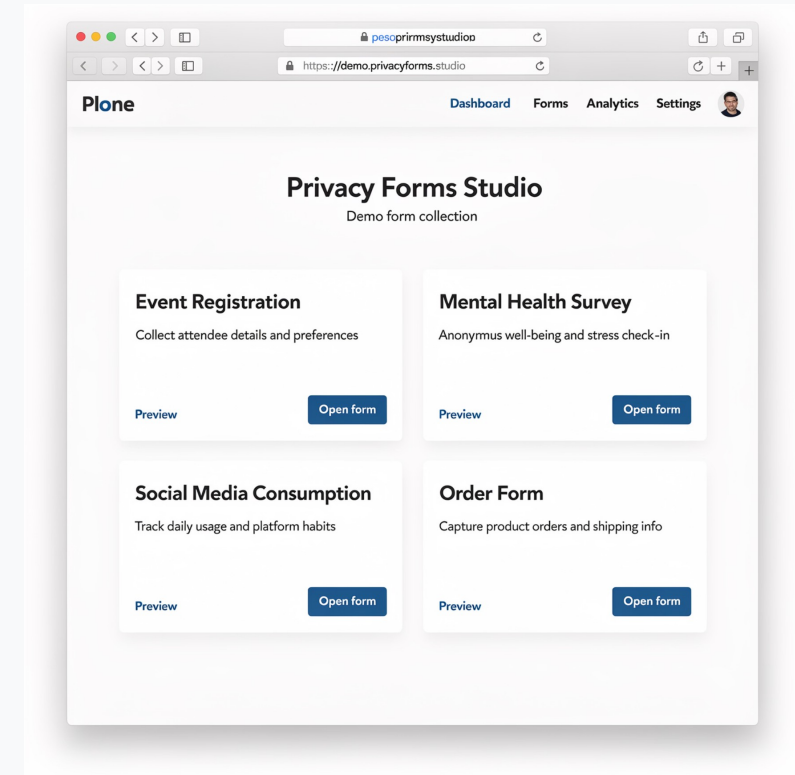
Demo Forms

10

Languages

LTR & RTL

All 10 languages covered: **English, Spanish, French, German, Arabic, Chinese, Japanese, Portuguese, Russian, Hindi** — spanning both left-to-right and right-to-left scripts.



demo.privacyforms.studio

Questions & Discussion

Thank you for your attention!

privacyforms.studio

Product website

hello@privacyforms.studio

Get in touch

docs.privacyforms.studio

Documentation

github.com/zopyx/zopyx.surveyjs

Source code