UnFold:
A Modular Microtransaction System for Publishers

Sweyn Venderbush

Advisor: Ruzica Piskac

E-mail: sweyn.venderbush@yale.edu

Overview

UnFold is a framework for allowing publishers to charge users for each article they wish to read. A user will be able to pay a small amount ($0.01-$1) to unlock access to a specific article forever. The framework will allow for cross-site functionality, so any site can collect payments from a user via the user’s central account to minimize transaction friction. The user will similarly be able to manage all of their articles as well as their account balance from a central portal.

Motivation

Publishers are finding it increasingly difficult to monetize their content. The rise of the internet resulted in a massive drop in print newspaper sales. The flood of free web content also forced many publishers to attempt to publish much or all of their content online for free, supported by ads. It is clear that for most publishers, this model isn’t sustainable. The
rise of ad-blocking software used by readers has resulted in a sharp decline in revenue. Even before the explosion of ad-block, publisher saw that ad revenue wasn’t sufficient to sustain publication the way newspaper sales had been.

As a result of these two factors, many newspapers (notably the New York Times and Wall Street Journal) moved to “paywall” structures, where users had to subscribe monthly to access the vast majority of content. This is far from a perfect solution. Unlike with a traditional newspaper, where the reader would make a selection of which paper to read and read it via delivery every day, readers on the web tend to be less loyal, consuming content from a wider variety of publications and finding articles to read via social media and web searches. The paywall model doesn’t lend itself well to this form of consumption, since subscribing to a large number of publications is too expensive, especially when the reader ends up only consuming a few articles per month per publication. As a result, paywalls often result in limiting access to publications without creating long term readers with allegiance to the publication.

Description of Functionality

The application will consist of 3 parts:

1. A backend server to manage transactions as well as track and return a user’s balance, list of owned articles, etc.

2. A server plugin for the publisher’s site to allow them to collect payments.

3. A consumer-facing front-end interface to manage your account.

Below is an example user flow to illustrate functionality:

- A publisher adds the UnFold plugin to their site and connects it to their publisher account.
• A user signs up for an UnFold account and adds a balance to their account.

• The user visits the publishers site wishing to read a specific article. On the webpage for that article, the user is presented with the option to purchase the article for a small fee determined by the publisher. The user accepts and this fee is deducted from their balance. If they didn’t have a sufficient balance to make the purchase, they would be given the option to refill their account before continuing.

• The user reads the article as they would normally. Since they now own the article, at any point if they come back to the article, they will be able to access it without having to re-pay.

**Technologies Utilized**

All backend functionality will be built in Python using the Django framework. Necessary frontend elements (client signup, article payment page) will be built using modern web standards in HTML5 and CSS3. The backend plugin for publishers will be built as a decorator in Python for Django.

The system will be built in a modular way to support the creation of additional “wallet” accounts that a user may use to pay a publisher. Just as OAuth allows many sites the option to accept login via Google, Twitter, and Facebook, this modular implementation will allow publishers to accept payments from users with accounts on UnFold, FooFold, BarFold, etc.

**Deliverables**

**Base Deliverables**

• Client site allowing account creation and ability to view balance and refill account (may not have a true mechanism to collect real world payments).
• Backend API that allows publishers to query for a user’s account information and execute transactions to allow a user to buy an article.

• Publisher-side plug-in for Django that allows a site to execute payment collection as well as accompanying front-end to allow interstitial page to accept payments before article is shown.

• Paper analyzing the need for this sort of solution in the publishing space as well as analysis of the success of UpFold as a proof of concept.

Reach Deliverables

• Implement Stripe API to allow collection of real world payments.

• Implement payment collection in a modular way to allow for easy extension so publishers may collect payments using a variety of services all built on top of the UnFold framework.

• Allow for preview of article to be shown before payment is asked for/collected.

• Allow different types of access to articles, including access to an entire portion of a site with one payment (e.g., today’s paper) or for a limited period of time (e.g., the entire site for one week).

• Allow for item tracking in a more sophisticated way than by URL.