External Project

Project Title: Secure Client-Side Backup and Telemetry Pipeline for Browser Extensions

Company: OpuLnt Technology LLC

Author(s): Prateek De and OpuLnt Technology LLC

Version: 1.0

Date: October 15, 2025

**CONFIDENTIALITY NOTICE: This document contains proprietary information belonging to Prateek De and OpuLnt Technology LLC. It may not be reproduced, distributed, or used without explicit written permission. All ideas and technical details contained herein are protected intellectual property.**

# Executive Summary

This project introduces a secure, client-side encrypted backup and telemetry pipeline for browser extensions, enabling users to safely back up and restore personal autofill data without exposing plaintext information to the cloud. The system uses a Safari extension to manage local data, Firebase Auth for identity, Firestore as encrypted storage, and Cloud Functions to aggregate anonymous telemetry for public synonym packs and accuracy improvements.

# Invention Overview

Problem: Existing browser autofill systems either store data insecurely or rely on server-side encryption, creating privacy risks. Additionally, they lack an effective way to crowdsource anonymous label-matching improvements while preserving privacy.  
  
Solution: This invention provides a client-side encrypted pipeline where data is encrypted locally before backup, stored securely in Firestore, and decrypted only on the client. Anonymous telemetry and public synonym packs enable the system to get smarter collectively without ever leaking personal data.

# Technical Architecture

The architecture consists of five main components:  
  
1. Safari Extension (Client): Handles form field storage, encryption, decryption, and interaction. Includes IndexedDB, Crypto Engine, and Options UI.  
2. Firebase Auth: Provides unique user identity through anonymous or federated sign-in, assigning a UID for Firestore access.  
3. Firestore: Acts as encrypted storage, with /vault for encrypted user data, /events for telemetry, and /public for synonym packs.  
4. Cloud Functions: Runs scheduled aggregations on telemetry data and publishes non-PII statistics to /public.  
5. Public Data: Holds shared synonym packs and domain statistics, improving accuracy for all users.  
  
Pipeline Flow (textual diagram):  
Safari Extension → Firebase Auth → Firestore (encrypted writes)  
Safari Extension ← Firestore (encrypted reads)  
Safari Extension → Public Data (read-only packs)  
Firestore → Cloud Functions → Public Data (telemetry aggregation)

# Detailed Technical Description

Encryption: The client derives a 256-bit key from a user passphrase using PBKDF2 or Argon2id with a local salt. Each record is encrypted with AES-GCM, generating a ciphertext and IV. Plaintext data never leaves the client.  
  
Backup Flow: When triggered, the extension signs in to Firebase anonymously, derives the key, encrypts each local record, and uploads ciphertext to /users/{uid}/vault. Firestore Rules ensure that only the owner can write within their UID scope.  
  
Restore Flow: The extension signs in, fetches encrypted docs from Firestore, decrypts them locally, and updates IndexedDB based on timestamps (Last-Write-Wins policy).  
  
Telemetry: Non-PII events (like label match confidence) are written to /users/{uid}/events. Cloud Functions aggregate these periodically to produce domain-level correction statistics, stored under /public/domainStats.  
  
Synonym Packs: Public synonym packs are stored at /public/synonymPacks and read by extensions to improve label matching without exposing user data.

# Implementation Calendar

Week 1: Environment setup — extension scaffold, Firebase connection, IndexedDB.  
Week 2: Local data model + crypto — PBKDF2/Argon2id key derivation, AES-GCM encryption.  
Week 3: Backup & restore flows — encrypt/upload and fetch/decrypt implemented end-to-end.  
Week 4: Security rules + telemetry — Firestore rules, event logging, Cloud Functions setup.  
Week 5: Public packs + polish — shared synonym packs, telemetry aggregation, UI refinements.  
Week 6: Documentation and testing — finalize diagrams, write README, end-to-end testing.

# Use Cases

1. Backup: A user clicks 'Backup', enters their passphrase, and their form data is encrypted and stored securely in Firestore.  
2. Restore: On a new device, the user installs the extension, signs in, enters their passphrase, and their data is restored locally.  
3. Telemetry: The extension logs anonymous correction rates, which improve global synonym packs without collecting PII.  
4. Public Packs: The extension reads synonym packs from /public to better understand form labels.

# Intellectual Property & Ownership

This project and all related intellectual property are owned by Prateek De and OpuLnt Technology LLC. All rights are reserved. Unauthorized reproduction or distribution of this document or the concepts contained herein is strictly prohibited. This document is intended as a theoretical and technical disclosure for IP protection and potential patent filing.

# Signatures

Authorized Signatory (OpuLnt Technology LLC): \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_

Prateek De : Prateek De Date: 15th Oct, 2025