

Designing Out SNAP Fraud

Han Ngo

Independent researcher

<https://truonghan.com>

October 9, 2026

Abstract

The Supplemental Nutrition Assistance Program (SNAP) cost about \$100 billion in federal funds in fiscal year 2024 and served an average of 41.7 million people each month. Public debate often treats the program’s payment error rate, near 11 percent in recent years, as a fraud rate. That reading is wrong. The error rate counts overpayments and underpayments from all causes, and most of it reflects mistakes by households and state agencies. The most studied fraud, trafficking (selling benefits for cash), diverted an estimated 1.6 to 2.0 percent of benefits from 2015 to 2017, almost all of it through small stores. Card skimming, a newer threat, steals from recipients. This report builds a taxonomy of SNAP fraud and explains why trafficking persists: the payment record carries no item list, and the shopper and the cashier both profit from the lie. It proposes a layered architecture covering eligibility, credentials, transactions, merchant payouts, and analytics. Its central problem is finding price anomalies across millions of daily transactions without human basket review. The answer is store-level scoring, automated graduated responses, and an investigation queue capped by budget. A simulation on synthetic data finds that price features alone miss disguised trafficking, while reconciling sales against supplier purchases still reaches stores holding about 60 percent of trafficked dollars when investigations are perfect and invoices honest. Under more realistic assumptions, with imperfect investigations, labels only from reviewed stores, some forged invoices, and a mix of disguises, capture falls to 38.6 percent, though review still pays for itself. The report also reviews related work and closes with harms, a policy alternative, and a phased roadmap.

Keywords: SNAP; EBT; benefit trafficking; card skimming; program integrity; price anomaly detection.

Licensed under CC BY 4.0. Interactive version: <https://truonghan.com/designing-out-snap-fraud/>.

1 Introduction

SNAP is the largest food assistance program in the United States. The US Department of Agriculture (USDA) funds the benefits and sets the rules. States determine eligibility, issue benefits, and run their own integrity units. USDA’s Food and Nutrition Service (FNS) authorizes and polices the retailers that accept benefits. In 2026 USDA renamed FNS the Food and Nutrition Administration (FNA) [42]. This report keeps the name FNS because the studies and regulations it cites use that name.

Households receive benefits on an Electronic Benefit Transfer (EBT) card. The card works like a debit card restricted to a closed network of authorized retailers. A state loads the monthly benefit onto the household’s account. The household buys eligible food at an authorized store, and the store’s EBT processor debits the account. Within a few banking days the store receives the same

amount in its bank account. Most EBT cards issued until recently used a magnetic stripe and a PIN. They carried no chip.

The scale is large on both sides of the transaction. USDA’s Economic Research Service reports that SNAP served a monthly average of 41.7 million people in fiscal year 2024, about 12 percent of the US population, at a federal cost of \$99.8 billion [34]. Federal spending rose to \$101.7 billion in fiscal year 2025, and participation averaged 42.1 million people per month [25]. About 265,000 retailers held authorization that year [35]. Superstores and supermarkets make up a small share of those stores but redeem most of the benefits. In fiscal year 2021 they were 15 percent of authorized firms and redeemed over 79 percent of benefits [38]. Convenience stores, small groceries, and specialty shops make up most of the authorized locations but redeem a small share of the dollars. Convenience stores alone were 44 percent of firms and 5 percent of redemptions that year [38].

Fraud matters for two reasons. The first is budgetary. A benefit dollar that becomes 50 cents of cash for a recipient and 50 cents of profit for a store buys no food. The second is legitimacy. SNAP depends on public support, and that support erodes when voters believe the program leaks. Inflated claims about fraud do their own damage. They lead to controls that cost more than they save and that push eligible families out of the program. A sound design starts from accurate numbers (Figure 1).

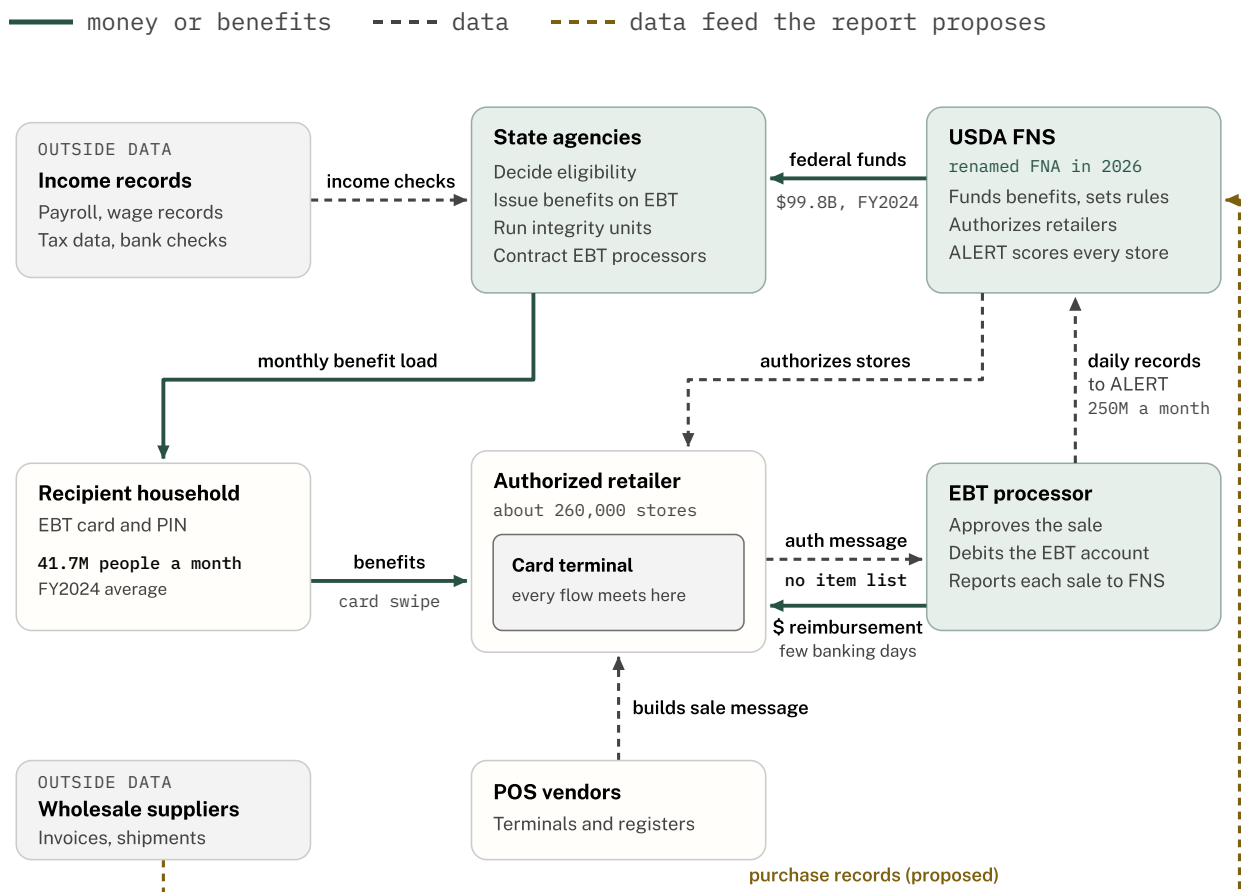
Contributions. This paper makes five contributions.

1. A taxonomy of SNAP fraud that separates fraud from improper payments. It matches each fraud type to the party that commits it, the party that loses, and the control that fits (Section 3).
2. A layered program and software architecture covering eligibility, credentials, transactions, merchant payouts, and analytics. It rests on evidence that a colluding shopper and store cannot shape (Sections 6 and 7).
3. A method for price anomaly detection that scores store-months instead of baskets, shrinks small-store scores toward their peers, ranks stores by expected loss, and caps human review at a queue sized to the investigator budget (Section 8).
4. A reproducible simulation study, with code, that tests the scoring and ranking method (Section 9). The data are synthetic, because EBT records today carry no item data. Against disguised trafficking, price features alone catch between 0 and 19.2 percent of trafficked dollars, while supply reconciliation keeps capture near 60 percent. A realistic scenario with imperfect investigations, labels only from reviewed stores, forged invoices, and a mix of disguises cuts capture to 38.6 percent and precision to 75.7 percent, though net benefit stays positive. Expected-loss ranking ties probability-only ranking, shrinkage shows no measurable effect, and random audits protect honest stores without raising capture. The study tests the logic of the method. It does not estimate real fraud.
5. An analysis of the harms each control imposes on honest stores and households, and of the cash-out policy alternative (Sections 10 and 11).

2 Related work

This paper draws on seven bodies of work. Each covers part of the problem. None combines a collusion-resistant program design with budget-capped review of item-level prices in a food benefit program.

SNAP trafficking measurement and enforcement. Most knowledge of SNAP trafficking comes from government studies, not academic journals. FNS publishes a series of studies that estimate the trafficking rate from investigation outcomes and from stores that its ALERT system



weak point The authorization message carries store, card, amount, date, and time. No item list reaches FNS, so every check infers fraud from amounts and timing.

Figure 1: Who does what in SNAP. Federal funds, state eligibility, and private payment rails all meet at one place: the store's card terminal. **Read it as:** many actors touch each benefit dollar, yet the one record they all share holds an amount and no basket. The amber feed is the report's proposal: purchase records the store does not write.

flags [50]. GAO has questioned the assumptions behind those estimates [48]. An earlier GAO report urged FNS to define the stores most likely to traffic and to aim its limited monitoring at them [46]. USDA's Office of Inspector General analyzed 21 months of retailer and transaction data. Among other problems, it found authorized retailers whose owners used the Social Security numbers of deceased people [40]. CRS reports summarize the enforcement record and the limits on acting against recipients [10, 11]. This paper takes its numbers from these sources. Its aim differs. Those studies measure the problem and audit the current program. This paper proposes the data and controls for a new one.

Statistical fraud and anomaly detection. Bolton and Hand reviewed statistical fraud detection for money laundering, credit card fraud, telecommunications fraud, and computer intrusion [4]. They note that fraudsters adapt and in time get around fixed controls, so detection has to back up prevention. Chandola, Banerjee, and Kumar survey anomaly detection and group the methods by the assumption each makes about what normal data look like [7]. Dal Pozzolo and coauthors

model credit card fraud detection under realistic conditions [15]. Investigators can check only a small set of transactions in time, so labels arrive late and only for the cases someone chose to check. The store-month scoring in Section 8 belongs to this tradition. It differs in two ways. It scores a merchant over a month, because a single trafficking swipe looks normal. And it adds evidence from suppliers, which a card-side detector never sees.

Fraud analytics in health care and public programs. Health insurance faces a similar problem: many claims, few auditors, and providers who control what they bill. Li and coauthors survey statistical methods for health care fraud detection [29]. Kumar, Ghani, and Mei built a system that predicts which health insurance claims contain payment errors, so that auditors can fix the riskiest claims before payment [28]. Medicare’s Fraud Prevention System analyzes claims data and generates leads for program integrity contractors. GAO found that 22 percent of the Medicare fraud investigations those contractors opened in fiscal year 2016 began from its leads [47]. SNAP differs in one structural way. A Medicare claim lists the services billed. An EBT record lists only an amount. The transaction layer in Section 7.3 exists to close that gap.

Audit selection under a budget. Elkan showed that a classifier’s probability estimates can drive cost-optimal decisions directly: act when the expected cost of acting is lower than the expected cost of not acting [18]. The review threshold in Section 8.4 applies that rule, with expected loss on one side and the cost of a review on the other. Tax agencies face the same allocation problem. Black and coauthors studied the models that inform IRS audit selection and showed that the choice of model objective changes how the audit burden falls across income levels [3]. Glaeser and coauthors used a public tournament to build models that predict restaurant hygiene violations from online reviews, so that Boston could target its inspectors [23]. This paper adds two things to that line of work. It fixes the queue length from the investigator budget before the run. And it ranks by expected dollars lost, not by the probability of a violation alone, though the simulation in Section 9 finds no difference between the two in its setting.

Shrinkage and empirical Bayes. James and Stein showed that when three or more means are estimated together, shrinking each estimate toward a common value lowers total squared error [27]. Efron and Morris recast the estimator in empirical Bayes terms and showed large gains on real data, including baseball batting averages from small samples [17]. A small store with few transactions poses a small-sample problem of this kind. Its raw excess-price share is noisy. Shrinking it toward the peer average is meant to keep one odd basket from pushing the store up the queue. The contribution here is the application, not the method, and the simulation in Section 9 finds no measurable effect from it.

Adversarial adaptation. Dalvi and coauthors framed classification as a game in which the adversary changes its behavior in response to the classifier [16]. Biggio and Roli review a decade of work on attacks against learned models, including evasion of a deployed model [2]. Trafficking stores are adversaries of this kind. Section 5.5 notes that they learn ALERT’s flags and stop using round amounts. Section 9 tests three such adaptations, including a store that keeps every price at shelf level and bills extra units instead. The architecture’s answer is evidence the store does not control, such as supplier invoices, rather than a better classifier alone.

Algorithmic harms in welfare fraud systems. Automated fraud detection in welfare programs has a record of serious failures. In Michigan, the MiDAS system issued automated fraud findings against unemployment claimants. The Sixth Circuit noted a state review that found 93 percent of the reviewed fraud determinations did not involve fraud [5]. In Australia, the Robodebt scheme raised debts against welfare recipients by averaging their annual income. A Royal Commission found it unlawful and called it a crude and cruel mechanism [33]. In the Netherlands, a court struck down the legislation behind the SyRI risk-profiling system in 2020, because it breached the right to private life under the European Convention on Human Rights [49]. Rotterdam’s audit

office found that the city’s welfare fraud risk model lacked clear ownership, gave too little attention to fairness and transparency, and could produce biased outcomes [32]. Eubanks documents how automated eligibility and risk systems in the United States fall hardest on poor households [19]. These cases share features this paper tries to avoid: findings without a human decision, opaque models, a burden placed on the accused to disprove a score, and targets drawn from the poorest participants. The rules in Section 10 respond to that record. Scores open cases and never decide them. Every automated action expires and can be appealed. And no household loses benefits without a person deciding the case.

3 A taxonomy of SNAP fraud

The Congressional Research Service groups SNAP integrity problems into trafficking, retailer application fraud, household errors and fraud, state agency errors and fraud, and external scams that steal benefits [10]. This report uses a slightly finer grouping because each type calls for a different control (Figure 2).

Table 3.1. A taxonomy of SNAP fraud

Type	Who commits it	Who loses	Mechanism	Main control today
Trafficking	Recipient and store, together	Taxpayers	Card swiped for a sale that does not happen; store pays recipient cash at a discount	ALERT transaction analytics, undercover buys, retailer disqualification
Recipient eligibility fraud	Applicant	Taxpayers	Underreported income, invented household members, hidden assets, enrollment in two states	State verification, data matching, quality control reviews, intentional program violation hearings
Retailer ineligible sales and overcharging	Store, sometimes with shopper	Taxpayers or recipient	Benefits used for alcohol, tobacco, or household goods; prices inflated for EBT shoppers	Undercover buys, complaints, term disqualification
Card skimming and cloning	Outside criminals	Recipient	Device copies magstripe data and PIN; clone drains the account soon after the monthly load	Card replacement policies, chip cards (new), recipient card locks
Insider fraud	Caseworker or state employee	Taxpayers	Fictitious cases, benefits issued to accomplices, data sold	Audits, separation of duties, access logs
Organized rings	Groups spanning stores, recruiters, and card holders	Taxpayers or recipients	Many cards trafficked through a few stores; mass skimming operations	Joint federal and state investigations, network analysis

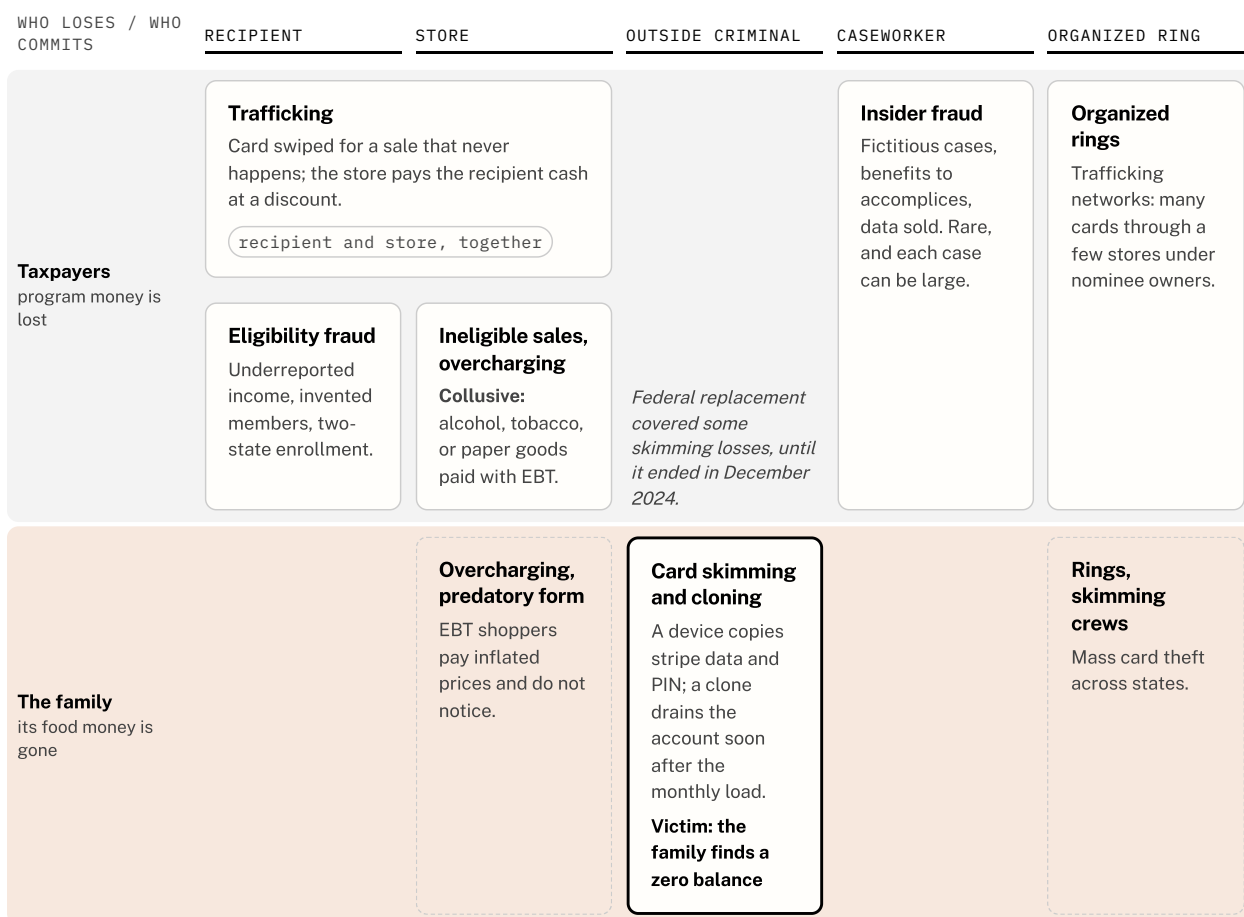


Figure 2: Six kinds of SNAP fraud: who commits it, who pays. Most fraud drains taxpayer money; skimming steals the family's food money. **Read it as:** five types sit in the taxpayer lane. Skimming sits only in the family's lane. Overcharging and rings reach the family in some variants (dashed cards).

Trafficking. Trafficking is the exchange of benefits for cash or anything other than eligible food. The typical case involves a recipient who lets a store clerk swipe the card for an amount, say \$100, and receives a smaller amount of cash. Federal prosecutors have described stores paying about 50 cents on the dollar in some cases [43, 44] and 50 to 65 cents in another [45]. Rates vary by case. Trafficking requires a retailer in almost every case, because only an authorized retailer can turn an EBT debit into a deposit. A recipient can also traffic by selling the card and PIN to a third party, who then shops with it.

Recipient eligibility fraud. An applicant may underreport earnings, omit a working household member, add a person who does not live in the home, or enroll in a second state. Honest mistakes produce the same outcomes, so eligibility problems show up in the error rate and only some of them are fraud.

Retailer ineligible sales and overcharging. A store may let EBT pay for items the law excludes, such as alcohol, tobacco, or paper goods. A store may also charge EBT shoppers more than cash shoppers, or inflate the price of real food items. The second practice can be collusive (the shopper accepts inflated prices in exchange for credit or cash) or predatory (the shopper does not notice).

Card skimming and cloning. Skimming is theft from recipients. A criminal attaches a

device to a store’s card reader or PIN pad. The device records the magnetic stripe data and the PIN. The criminal writes the data onto a blank card and drains the account, often within hours of the monthly benefit load [12]. The recipient arrives at the store to find a zero balance. Magnetic stripe cards are easy to clone because the stripe carries static data that works on every swipe. A chip card generates a one-time code for each purchase, so copied data does not work for a second purchase.

Insider fraud. State employees with system access can create fictitious cases or issue benefits to accomplices. Cases are rare, and each can be large because the insider controls the record.

Organized rings. Rings combine the other types, for example by running several trafficking stores under nominee owners or skimming crews across states. They leave patterns across many cards and stores that no single transaction shows.

3.1 Fraud versus improper payments

The claim that “11 percent of SNAP is fraud” comes from the national payment error rate. USDA computes that rate each year from a quality control sample of cases [37, 41]. Reviewers check whether each sampled household received the correct benefit. Any difference above a small tolerance counts as an error, in either direction.

Table 3.2. National payment error rates by fiscal year

Fiscal year	National payment error rate	Overpayment rate	Underpayment rate
2023	11.68%	10.03%	1.64%
2024	10.93%	9.26%	1.67%
2025	10.62%	9.28%	1.34%
			derived

Four facts separate this rate from fraud. First, it includes underpayments, which are money the program failed to pay to eligible families. Second, it counts errors regardless of cause, and the largest causes are mistakes in income and household data made by households or caseworkers. USDA’s own release for fiscal year 2024 stated that the rate is not a measure of fraud [39]. Third, it measures eligibility and benefit amounts only. It does not measure trafficking at all, because a trafficked benefit was correctly issued to an eligible household before it was sold. Fourth, error rates move with administrative load. They fell to about 3 percent around 2013 [11], then rose after the pandemic disrupted state operations and staffing (Figure 3).

The error rate is still a serious problem. Improper payments at a rate near 11 percent of benefits are real money [37], and Congress has tied state cost sharing to error rates starting with fiscal year 2028. From that year a state with an error rate of 6 percent or more pays 5, 10, or 15 percent of benefit costs, depending on the rate [31]. The point is narrower: the error rate and the fraud rate measure different things, and controls aimed at one do little for the other. The best available national fraud measure is the trafficking estimate. FNS’s eighth study in that series, covering 2015 to 2017 and published in 2021, estimated that 1.6 percent of benefits were trafficked under its updated definition (about \$1.0 billion a year) and 2.0 percent under the older definition (about \$1.3 billion a year) [50]. The earlier 2012 to 2014 study put the rate near 1.5 percent [11]. GAO has warned that the true figure is uncertain and could range from about \$960 million to \$4.7 billion a year, because the estimate rests on untested assumptions about how much of a trafficking store’s volume is trafficked [48].

All SNAP benefits, about \$100 billion a year (FY2024) = 100%



Outlined: the 0 to 12% slice, enlarged below on one scale.

0% 2% 4% 6% 8% 10% 12%

Payment error rate FY2024, all causes, both directions **10.93%** about \$10B



■ Overpayments **9.26%**: mostly income and household data mistakes

■ Underpayments **1.67%**: money families were owed and not paid

Trafficking (fraud) 2015 to 2017 estimate **1.6% to 2.0%**



■ **1.6%** updated definition, about \$1.0B a year ▨ to **2.0%** older definition, about \$1.3B a year

Errors, either direction

Any gap above a small tolerance counts, whatever the cause. USDA, FY2024: the rate is not a measure of fraud.

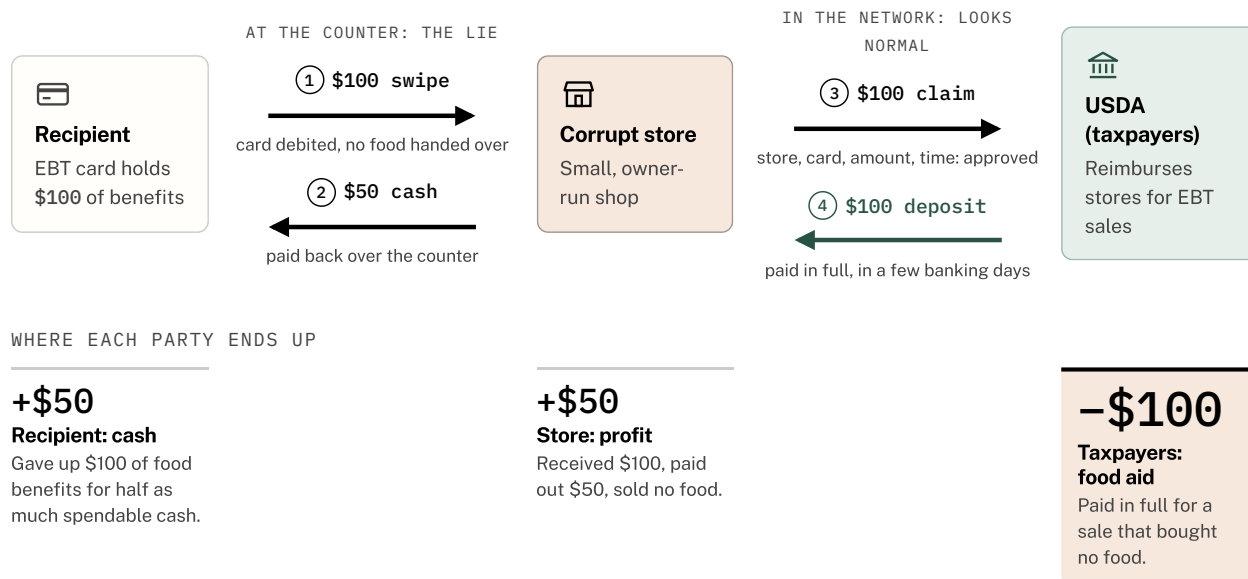
Trafficking is not in the 11%

A trafficked benefit was correctly issued to an eligible household, then sold for cash. The error rate never sees it.

Read sizes, not sums

Different years, methods, and measures. GAO: true trafficking is uncertain, about \$960 million to \$4.7 billion a year.

Figure 3: The 11 percent is mostly error, not fraud. The payment error rate counts mistakes in both directions; trafficking, the best fraud measure, is about 1.6 to 2.0 percent. **Read it as:** compare bar lengths. The grey bar is error, mostly honest mistakes; the red bar is the fraud estimate, under a fifth of its length. Controls aimed at one do little for the other.



Illustrative Example amounts. Federal cases describe rates around 50 cents on the dollar; some report higher or lower.

Figure 4: How one trafficking swipe moves the money. The recipient and the store split \$100 of food aid as cash; the program pays in full and no food changes hands. **Read it as:** both people at the counter gain, so neither complains. The full loss lands on the program, which sees only a normal approved sale.

4 Anatomy of trafficking

The money flow of a single trafficking transaction is simple (Figure 4).

The core weakness. The EBT authorization message carries the store, the card, the amount, the date, and the time. It carries no list of items. FNS’s 2016 feasibility study on capturing purchases at the point of sale confirmed that only total transaction amounts reach FNS [36]. A trafficking store therefore never has to invent products. It keys in an amount, the network approves it, and the deposit follows. Every downstream check has to infer the fraud from the shape of the amounts and timings, because the record contains nothing else (Figure 5).

Both parties lie together. In most fraud, a victim has an incentive to report. In trafficking, the shopper and the cashier both gain. The recipient converts restricted benefits into cash, which can pay rent or utilities, or in worse cases buy drugs. The store keeps the spread. Neither party will complain, and any data either party enters can be shaped to look normal. This collusion property drives most of the design choices later in this report.

Why small stores dominate. The 2015 to 2017 study found that small stores (small and medium groceries and convenience stores) handled about 15 percent of redemptions but accounted for over 95 percent of trafficked dollars, and 99 percent under the updated definition [50]. Several factors explain the concentration. A small store is often owner-operated, so one person controls the register, the cash drawer, and the books. Large chains run centralized point-of-sale systems with item scanning, cashier logins, and loss-prevention audits, and a cashier who trafficked would be stealing from the employer. Small stores also have few transactions per day, so a trafficking swipe does not need to hide in a long queue of scanned baskets. FNS analyzes transactions of all store types but leaves large stores off its Watch List, because officials said those stores already have antifraud measures against trafficking [48].

The overcharging variant. A store can also traffic partially. The shopper buys real groceries

WHAT THE RECORD CARRIES

EBT AUTHORIZATION		APPROVED
STORE	retailer 0418823	
CARD	 5207	
AMOUNT	\$87.43	
DATE	day of sale	
TIME	14:02	

Sent to FNS daily, for every SNAP sale.

Illustrative values

WHAT IT LACKS: THE ITEM LIST

ITEM	QTY	PRICE
?	?	?
?	?	?
?	?	?
...		

NOT SENT

Only total amounts reach FNS (FNS 2016 feasibility study).

① **Store keys in an amount**
no basket behind it

② **Network approves**
valid card, enough balance

③ **Deposit follows**
store paid in full

Variant: overcharging (partial trafficking)

Real groceries go home, but the register charges more than they cost.

\$40 real groceries

\$15 shopper

\$15 store

Register charges \$70

Illustrative

The basket looks normal in size and timing, so amount-based analytics struggle to catch it.

Figure 5: The blind record: an amount with no basket. The EBT record says where, which card, how much, and when, but never what was bought. **Read it as:** a real \$87.43 basket and a trafficked \$87.43 swipe produce the same record. Every later check must guess from amounts and timing.

worth \$40, the register charges \$70, and the shopper receives \$15 in cash or store credit. The store earns \$15 above its margin. This version leaves a basket that looks normal in size and timing, so it is harder for amount-based analytics to catch. It also has a predatory form, where the store inflates prices for EBT shoppers who do not check their receipts.

The economics. Any restricted currency trades at a discount to cash. A dollar of benefits that can only buy groceries is worth less than a dollar to a household that needs to pay rent. In standard theory, a household that already spends more than its benefit on food treats the benefit like cash, so trafficking offers it little [26]. Retail panel data are less tidy: households spend 50 to 60 cents of each benefit dollar on SNAP-eligible food, far more than out of cash, even when the benefit is below their food spending [24]. A household with an urgent cash need, or one whose benefit exceeds its food budget, may value a benefit dollar at well under a dollar. Trafficking monetizes that gap. A store that can offer cash at 50 to 65 cents per dollar, the range prosecutors have described [44, 45], captures the remainder. Enforcement leaves the gap in place and raises the price of crossing it, by adding the risk of disqualification and prosecution to the store’s side of the trade.

5 Current detection and enforcement

5.1 Transaction analytics: ALERT

FNS’s main detection tool for retailer trafficking is the Anti-Fraud Locator using EBT Retailer Transactions (ALERT). State EBT processors send FNS daily records of every SNAP transaction. GAO reported in 2018 that ALERT scanned about 250 million transactions per month [48]. The system assigns each store a numeric score for the likelihood of trafficking. Stores above a threshold join a watch list, and analysts prioritize them using factors such as average transaction size relative to store type. Analysts also compare a suspect store with similar stores in the same ZIP code. CRS reported that over 80 percent of the retailer trafficking FNS detects is found mainly through EBT transaction analysis [11] (Figure 6).

FNS charge letters and agency decisions describe the patterns that drive these cases:

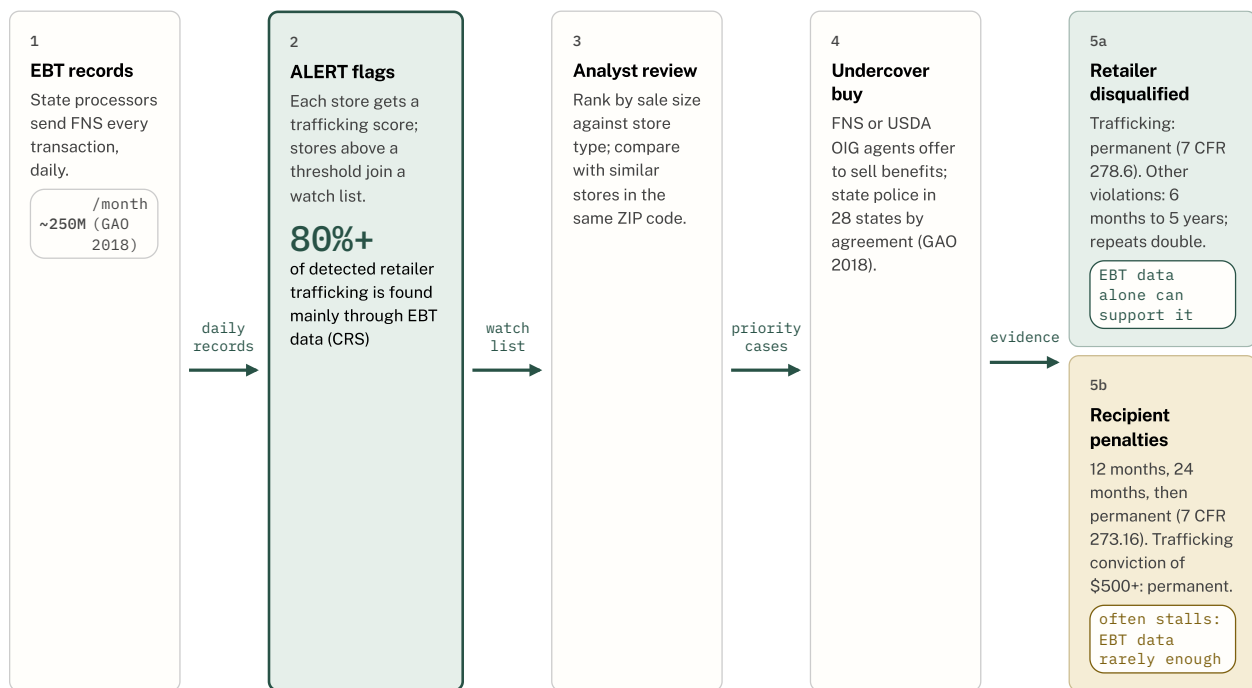
- A large number of transactions in repeated or round dollar values, such as many sales ending in .00.
- Multiple transactions from one household account within a short window, a pattern unusual for a small store with limited stock.
- Transactions that are excessively large for the store’s size and type.
- Multiple households making similar transactions within a short period.
- Accounts drained in one or two swipes, sometimes by out-of-state cards.

Each pattern is a proxy for “no real basket behind this amount.” Round totals are rare when real prices are summed. Repeat swipes from one card suggest the store is splitting a large trafficking amount to avoid a size flag. Large totals at a store with a few shelves of stock suggest the store sells more food than it holds.

5.2 Undercover investigations and penalties

FNS investigators and USDA’s Office of Inspector General conduct undercover buys, in which an investigator with a test card offers to sell benefits or to buy ineligible items. Under state law enforcement bureau agreements, which GAO counted in 28 states in 2018, state and local police also run their own undercover cases with SNAP cards [48].

Penalties are set by regulation. For retailers, 7 CFR 278.6 provides [9]:



Every stage runs after the sale: the store has already been paid.

Patterns that drive cases

Each is a proxy for "no real basket behind this amount."

- Many sales in round or repeated amounts, such as totals ending in .00
- Several swipes from one household account in a short window
- Sales too large for the store's size and type
- Many households making similar sales in a short period
- Accounts drained in one or two swipes, sometimes by out-of-state cards

Five structural limits

1. **Detects after the fact.** The store is already paid; owners can reopen under a new name.
2. **Sees only amounts.** A real \$87.43 basket and a trafficked \$87.43 swipe look the same.
3. **Targets stores, largely misses recipients.** EBT data alone is generally not enough to disqualify a recipient.
4. **Does not scale with investigators.** GAO: all stores reauthorized on a five-year cycle, regardless of risk.
5. **The card was weak.** Magnetic stripes made skimming cheap.

Figure 6: How SNAP catches trafficking today. Transaction analytics find most cases, people confirm them, and sanctions fall mostly on stores, after the money is paid. **Read it as:** the pipeline is a strong detector of crude store fraud. It acts late, sees only amounts, and rarely reaches the recipient side of a case.

Table 5.1. Retailer sanctions under 7 CFR 278.6

Violation	Retailer sanction
Trafficking	Permanent disqualification (278.6(e)(1))
Trafficking, with an effective compliance program in place before the violation and owners not involved	Trafficking civil money penalty in place of permanent disqualification (278.6(i))
Selling ineligible items after a warning, as a practice (expensive or conspicuous items, alcohol, tobacco)	5 years for a first sanction (278.6(e)(2))
Common nonfood items after a warning, or (e)(2) conduct without warning	3 years (278.6(e)(3))
Ineligible sales by owners or managers without a warning, or credit sales	1 year (278.6(e)(4))
Violations from carelessness or poor supervision	6 months (278.6(e)(5))
Repeat sanctions	The above periods double (278.6(e)(6))

The regulation lets FNS rely on “evidence obtained through a transaction report under an electronic benefit transfer system” [9]. That clause is what lets ALERT data alone support a permanent disqualification.

For recipients, 7 CFR 273.16 sets disqualification periods for an intentional program violation: 12 months for a first violation, 24 months for a second, and permanent for a third [8]. A court conviction for trafficking \$500 or more brings permanent disqualification on the first occasion. A fraudulent statement about identity or residence to receive benefits in more than one place brings a 10-year bar. Transactions involving firearms, ammunition, or explosives bring permanent disqualification. Transactions involving controlled substances bring 24 months, then permanent.

5.3 Eligibility and retailer standards

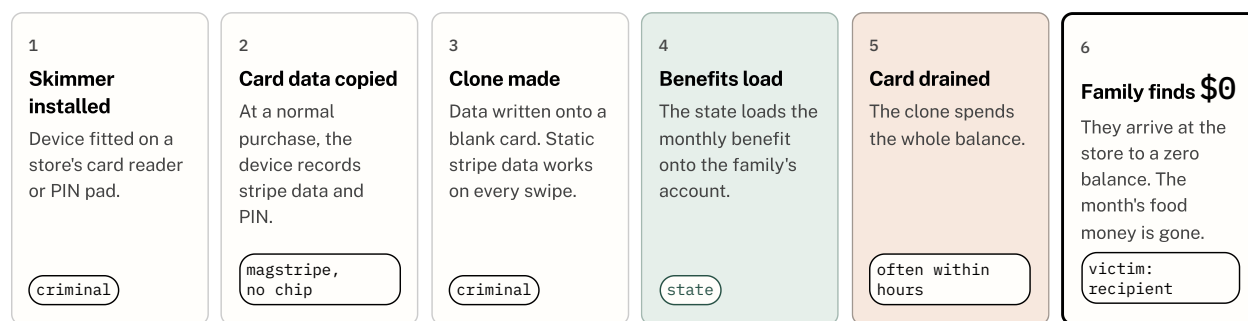
On the eligibility side, states verify income and identity using federal and state data sources. The 2018 Farm Bill required a National Accuracy Clearinghouse (NAC) to stop the same person from receiving benefits in two states at once. FNS issued an interim final rule in October 2022 requiring all state agencies to join the NAC and to act on its match results, with full compliance due by 2027 [21]. Before the NAC, states relied on the slower Public Assistance Reporting Information System (PARIS) for interstate matching.

On the retailer side, FNS published a final rule in December 2016 to raise stocking standards. It required stores qualifying on inventory to carry at least seven varieties in each of four staple food categories, with perishable items in at least three categories and at least three units of each variety [20]. The aim was partly integrity: a “store” with a few cans on a shelf and a busy EBT terminal is a classic trafficking front. Appropriations riders, beginning with section 765 of the Consolidated Appropriations Act of 2017, blocked enforcement of the variety and breadth provisions for years. USDA proposed updated standards in September 2025 and finalized them in May 2026. New authorizations are assessed against them 180 days after publication, in early November 2026, and existing stores at their next reauthorization [22].

5.4 Skimming response

The Consolidated Appropriations Act, 2023 (P.L. 117-328, Division HH, section 501) gave states temporary authority to replace stolen benefits with federal funds, limited to the lesser of the amount stolen or two months of benefits, and to two replacements per household per fiscal year [13]. A later continuing resolution extended the window to cover thefts through December 20, 2024.

THE ATTACK, ONE BENEFIT MONTH (NOT TO SCALE)



Why a chip fixes it: a magnetic stripe carries static data that works on every swipe. A chip card makes a one-time code per purchase, so copied data fails a second time.

WHO COVERS THE LOSS

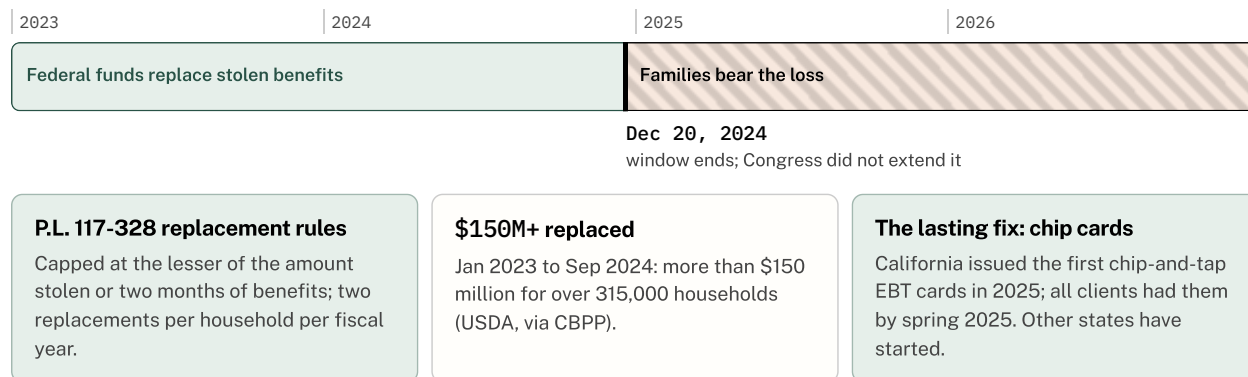


Figure 7: Skimming: how a family's benefits vanish. A copied magstripe card is drained soon after the monthly load, and since federal replacement ended in December 2024 the family bears the loss. **Read it as:** the theft hits the family, not the store or the program. Since December 2024 no federal backstop covers it, so the card itself has to change.

Congress did not extend it further [30]. According to USDA figures cited by the Center on Budget and Policy Priorities, states replaced more than \$150 million in stolen benefits for over 315,000 households between January 2023 and September 2024 [6]. The underlying fix is the card itself. California became the first state to issue chip-and-tap EBT cards in 2025, and state officials told legislators that all clients had received new cards by spring 2025 [14]. Other states have started similar migrations (Figure 7).

5.5 Limits of the current approach

The current model has five structural limits.

First, it detects after the fact. ALERT reads completed transactions, and the store has already been paid. By the time a store is disqualified, it may have trafficked for months, and the owners can reopen under a new name.

Second, it sees only amounts. Without items, ALERT cannot tell a real \$87.43 basket from a trafficked \$87.43 swipe. Smart traffickers learn the flags and stop using round amounts.

Third, it targets stores and largely misses recipients. EBT data alone is generally not enough to disqualify a recipient, so states struggle to act on the cardholder side of ALERT cases [11].

Fourth, it does not scale with investigator capacity. GAO found that FNS reauthorized all stores on a five-year cycle regardless of risk, and that its trafficking estimate rested on assumptions it had not tested [48].

Fifth, the card was weak. Magnetic stripe cards made skimming cheap. The program bore part of the loss through replacement, and recipients bore the rest after the replacement authority ended. Figures A1 to A4 in Appendix A model this response as UML operations diagrams.

6 Design principles

Four principles shape the architecture in the next section.

IMPORTANT

Collusion resistance. When the shopper and the cashier both cheat, any data they enter can be faked. A store can key fake item lines, and a shopper will confirm any receipt. The system therefore needs evidence that neither party controls. Examples include wholesale invoices from suppliers, distributor shipment records, the stock a field inspector sees on the shelf, and the behavior of the same cards at other stores. Third-party evidence is the only class of data that a colluding pair cannot shape (Figure 8).

IMPORTANT

Defense in depth. No single control stops all fraud types. Chip cards stop cloning and do nothing about trafficking. Item data constrains overcharging and only raises the cost of trafficking. Each layer should assume the layers before it have failed, and each should produce data the analytics layer can use to catch what got through.

IMPORTANT

Economic framing. Integrity spending is worth doing only up to the point where the next dollar of control saves at least a dollar of loss, counting the costs that controls impose on honest participants. Trafficking of about \$1 billion a year [50] is a large number in absolute terms and a small share of a \$100 billion program. A control that costs \$600 million a year to prevent \$300 million in trafficking is a bad trade.

IMPORTANT

Protect honest participants. False positives cut food to families. A wrongly disqualified store can leave a neighborhood with no place to use EBT. A wrongly flagged recipient can lose benefits during an appeal. Error costs fall on people with little cushion. Every automated action needs a reversible design, a fast appeal, and human review before any benefit cutoff.

7 A layered program and software architecture

The architecture has four transaction-path layers and one analytics layer that reads from all of them. Confirmed cases flow back into the analytics layer as labels (Figure 9).

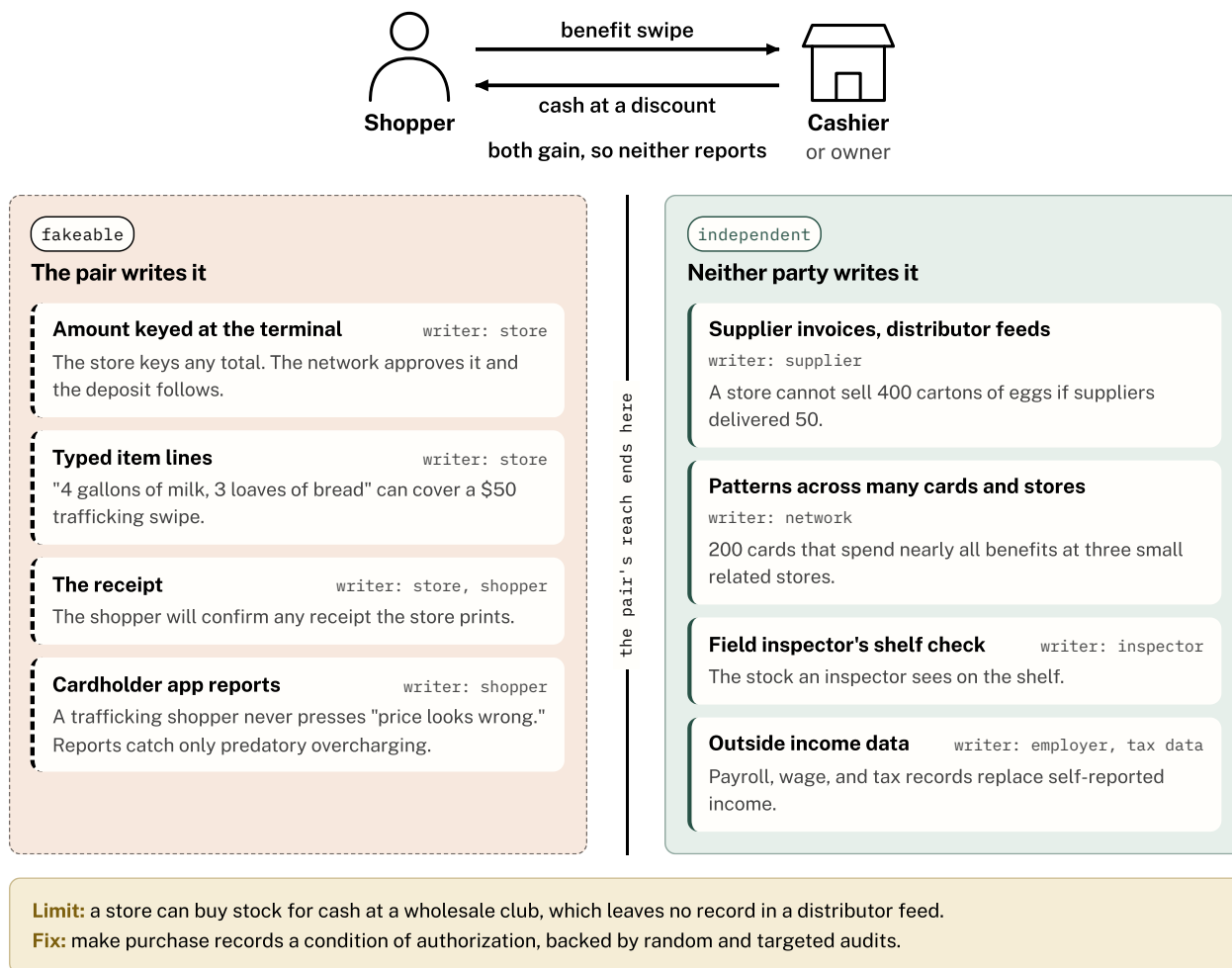


Figure 8: Collusion resistance: who writes the evidence?. Build detection on evidence the colluders cannot write. **Read it as:** every item on the left comes from the shopper or the store, so a colluding pair can make it look normal. Every item on the right comes from someone outside the deal.

The transaction-path layers act on each enrollment or sale. The analytics layer acts on aggregates over time, because most fraud signals are weak in a single event and strong in a pattern. Figures A5 to A7 in Appendix A model this architecture as UML software diagrams.

7.1 Eligibility layer

The eligibility layer decides who receives benefits and how much. Its job is to stop overpayments before the first payment, which is cheaper than recovering them later.

Automated income matching. At application and at recertification, the system should query payroll data (such as commercial employment verification services and state new-hire registries), state wage records, federal tax data where the law allows, and bank account verification. A discrepancy above a tolerance triggers a request for documents instead of a denial. The goal is to replace self-reported income with verified income where a source exists, and to save caseworker time for cases where no source exists, such as informal or gig work.

National duplicate-enrollment check. Every new application should check the NAC before

TRANSACTION PATH: ACTS ON EACH ENROLLMENT OR SALE

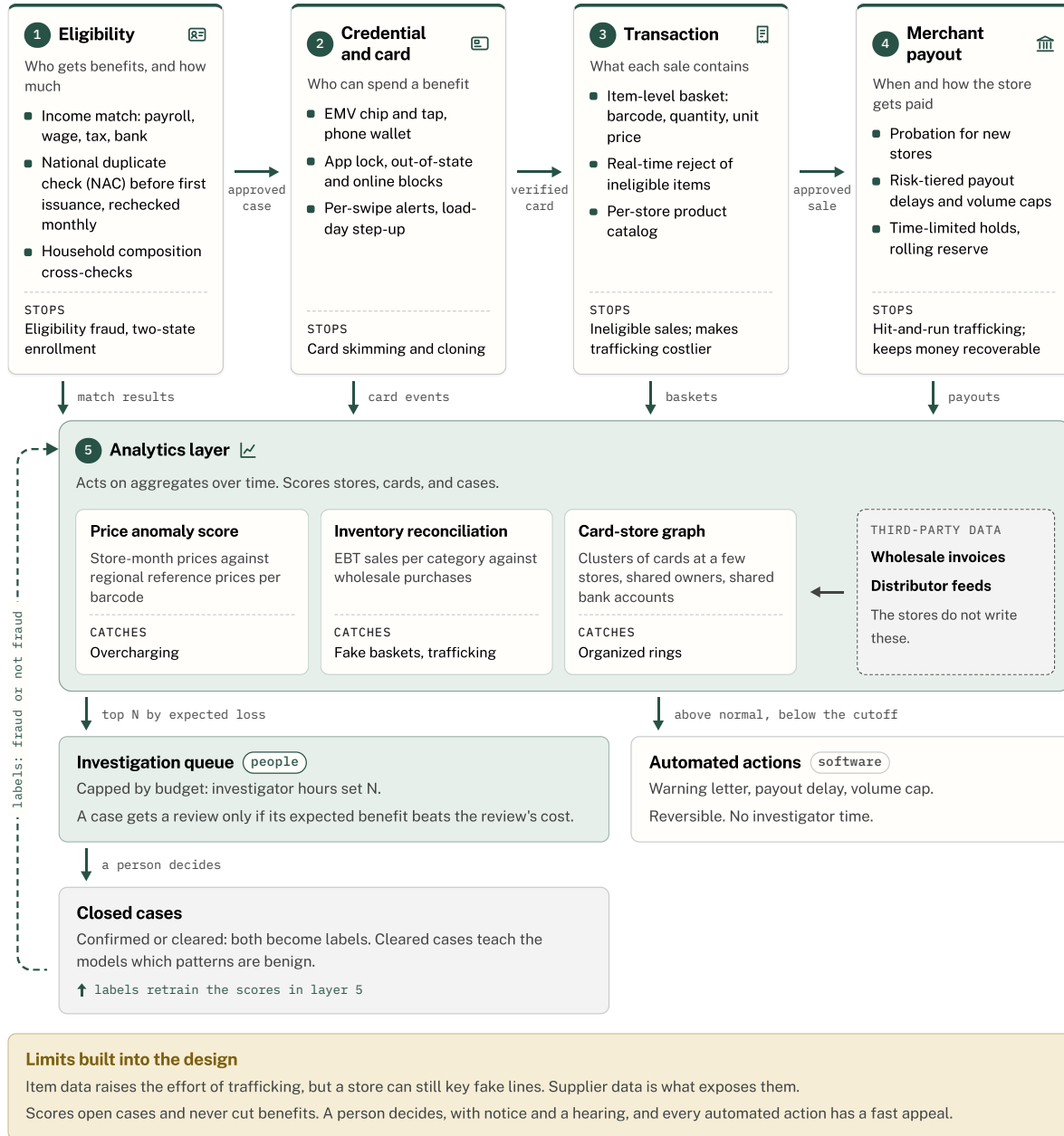


Figure 9: A layered program and software architecture. Four layers act on each enrollment or sale; one analytics layer reads them all, because fraud is weak in one event and strong in a pattern. **Read it as:** each layer assumes the one before it failed, and each feeds the analytics that catch what got through. People see only the top of a ranked list, and every closed case makes the next ranking better.

the first issuance, and every active case should be rechecked on a schedule, such as monthly. A match pauses the second state’s issuance and opens a resolution task with notice to the household. Many matches are benign, such as a family that moved and has not closed the old case. The software should treat a match as a question to resolve quickly, and it should close the old case when the household confirms the move.

The eligibility check runs at signup, at recertification, and on a schedule, and it settles every question before the first payment.

PSEUDOCODE

```
def check_eligibility(application):
    # Verified income replaces self-reported income
    # where a source exists.
    sources = [payroll_data, new_hire_registry, state_wage_records,
               federal_tax_data, # only where the law allows
               bank_account_verification]
    verified = [s.income_for(application.household)
                for s in sources if s.covers(application)]
    if not verified:
        # informal or gig work has no source
        assign_caseworker(application)
    for income in verified:
        if abs(income - application.reported_income) > INCOME_TOLERANCE:
            # a mismatch asks, never denies
            return request_documents(application)
    match = NAC.find_active_case(application.household)
    if match:
        pause_issuance(application)
        open_resolution_task(application, match, notify_household=True)
        # Most matches are moves; close the old case on confirmation.
        return "pending_resolution"
    return approve_first_payment(application)

def recheck_active_cases(active_cases): # on a schedule, such as monthly
    for case in active_cases:
        if NAC.find_active_case(case.household, excluding=case):
            pause_issuance(case)
            open_resolution_task(case, notify_household=True)
```

Household composition checks. Cross-checks against other benefit programs and state records can flag a member listed in two households. These checks catch fraud and honest error alike, so the response should be a request for clarification.

7.2 Credential layer

The credential layer controls who can spend a benefit. Its main target is theft from recipients.

Chip and contactless cards. EMV chip cards defeat magnetic stripe skimming, because a cloned stripe or a captured one-time code cannot authorize a second purchase. Tap-to-pay reduces exposure to tampered card slots. California’s rollout shows the main obstacle: retailers must upgrade terminals, and for a time most transactions still fell back to the stripe. The program should set a date after which stripe fallback is declined except at stores with an approved exception.

Phone wallets. Mobile wallet credentials add device binding and biometric unlock. The 2018 Farm Bill directed USDA to authorize mobile technologies for SNAP benefits and to approve up to five state demonstration projects before a nationwide rollout [1]. A wallet also gives the program a direct channel to the recipient.

Recipient app controls. A recipient app should let the household lock the card between uses, block out-of-state and online transactions by default, and receive an alert for every swipe. Locks and geography blocks stop most cloned-card drains. Per-swipe alerts let a household report theft within minutes.

Countering load-day drains. Skimming crews strike when the monthly benefit lands, because that is when the balance is highest. Two controls help. The first is to vary the load time within the day or to spread loads across days by case number, so a crew cannot predict the exact moment. The second is a step-up check for large spending soon after a load: if more than a set share of the balance is spent within a short window, at a store far from the recipient’s usual stores, the system asks for confirmation in the app before approving. The check must fail open for recipients without phones, or it will cut off the households least able to absorb a delay (Figure 10).

The credential check runs on every swipe and asks the household to confirm only a large spend soon after benefits load.

PSEUDOCODE

```
def authorize_swipe(card, txn, settings, now):
    # households report theft within minutes
    send_swipe_alert(card.household, txn)
    if txn.entry_mode == "stripe" and now >= STRIPE_FALLBACK_DEADLINE:
        if not txn.store.has_approved_exception:
            return decline("STRIPE_FALLBACK_ENDED")
    if settings.card_locked:
        return decline("CARD_LOCKED")
    if txn.is_out_of_state and settings.block_out_of_state: # default on
        return decline("OUT_OF_STATE_BLOCKED")
    if txn.is_online and settings.block_online: # default on
        return decline("ONLINE_BLOCKED")
    if needs_step_up(card, txn, now):
        if card.household.has_app:
            if not confirm_in_app(card.household, txn):
                return decline("NOT_CONFIRMED")
        # No phone means fail open, never a cutoff.
    return approve(txn)

def needs_step_up(card, txn, now):
    soon_after_load = now - card.last_load_time <= STEP_UP_WINDOW
    spent = card.spent_since_last_load + txn.amount
    large_share = spent > STEP_UP_SHARE * card.balance_at_last_load
    far_away = distance(txn.store, card.usual_stores) > STEP_UP_DISTANCE
    return soon_after_load and large_share and far_away
```

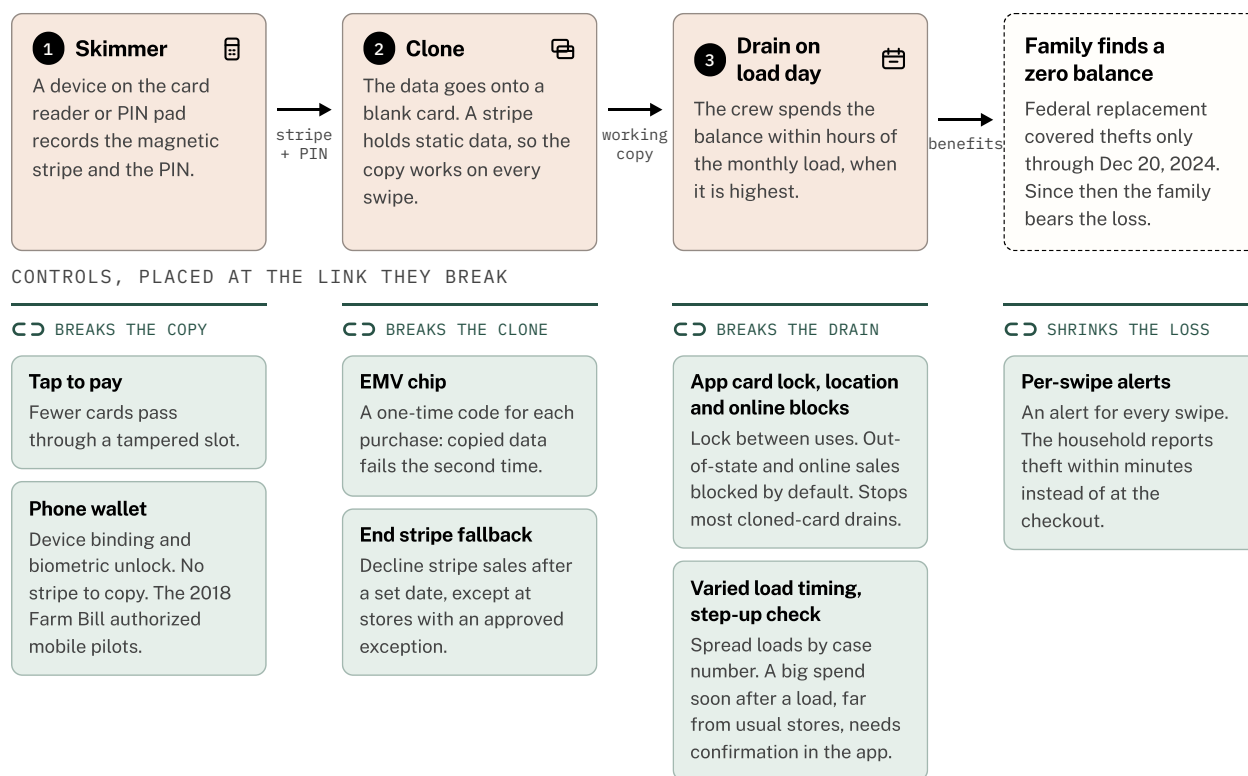
7.3 Transaction layer

The transaction layer is where the program can close the core weakness: the payment record with no item list.

Mandatory item-level basket data. Each EBT sale should carry a line for each item: the barcode (UPC or PLU), quantity, unit price, and SNAP-eligible flag. FNS’s 2016 feasibility study showed that item-level data can be captured from integrated electronic cash registers and matched to EBT transaction records at above 99 percent accuracy using card number, date, store, and amount [36]. The new standard would carry the basket in the authorization message itself or in a linked message sent within seconds.

Real-time rules. With item data in the message, the processor can reject a sale that includes

ATTACK PATH



Catch: in California's chip rollout, retailers had to upgrade terminals, and for a time most sales still fell back to the stripe. The step-up check must fail open for recipients without phones, or it cuts off the households least able to wait.

Figure 10: Hardening the card: breaking the skimming path. A skimming theft needs three steps to succeed, and each control breaks the chain at a different link. **Read it as:** the chip kills the clone at link 2, so it does the most work. Locks, blocks, and alerts catch whatever still gets through at link 3.

ineligible items before it completes. It can also reject or flag lines for products the store is not known to stock. The program would maintain a per-store product catalog, seeded from the store's own price list at authorization and from wholesale records over time. A convenience store that suddenly sells 30 pounds of beef in one basket, a product it has never bought from a supplier, would see the line flagged.

The processor validates the itemized basket inside the authorization and declines a bad one at the register with a reason code.

PSEUDOCODE

```
def validate_basket(txn, products, store_catalog):
    total = 0
    for line in txn.basket_lines:
        product = products.get(line.upc) # UPC or PLU
        if product is None:
            return reject(txn, "UNKNOWN_BARCODE", line)
        if not product.snap_eligible:
            return reject(txn, "INELIGIBLE_ITEM", line)
        # Catalog seeded from price list, then supplier records.
        if line.upc not in store_catalog[txn.store]:
```

```

        return reject(txn, "NOT_STOCKED", line)
        total += line.quantity * line.unit_price
    if total != txn.amount:
        return reject(txn, "TOTAL_MISMATCH")
    return accept(txn)

def reject(txn, reason_code, line=None):
    # Repeated unstocked lines are a signal for the monthly score.
    report_to_analytics(txn.store, reason_code, line)
    return Decline(txn, reason_code)

```

CAUTION

The limit. A corrupt store can type fake item lines. It can key “4 gallons of milk, 3 loaves of bread, 2 pounds of chicken” to cover a \$50 trafficking swipe. Item data therefore raises the effort of trafficking and gives analytics more to work with. It does not end the scheme on its own. Its value is that fake baskets must be consistent with the store’s purchases from its suppliers, which the store does not control. That link is what the analytics layer exploits.

Small-store capacity. Many small stores do not run integrated scanning registers. The standard needs a low-cost path, such as a subsidized, certified tablet point-of-sale app, or the rollout will drive stores out of the program.

7.4 Merchant payout layer

Card networks and payment processors face the same problem SNAP faces: some merchants are fronts for fraud. Processors such as Stripe and PayPal manage that risk at payout time. The SNAP payout layer should adopt the same tools.

Probation for new stores. A newly authorized store, or a store under new ownership, starts in a probation tier. It receives payouts on a longer delay, faces a monthly volume cap set from its stated size and stock, and gets an early field visit. Trafficking stores often show high volume soon after authorization, so probation targets the riskiest period.

Risk-tiered delays and caps. Every store sits in a tier set by its analytics score. Low-risk stores get the standard settlement time. Higher tiers get longer settlement delays and volume caps. A delay of a week or two does not stop an honest store from operating. It does let the program withhold funds when a case opens, which makes recovery possible (Figure 11).

Holds pending review. When a store’s score crosses a high threshold, the program can hold part of its payouts until review. A hold of this kind must have a time limit and a clear path to release, or it becomes a penalty without process.

Recovery reserve. For high-tier stores, a rolling reserve (a share of payouts held for a period) gives the program a fund to recover from after a disqualification.

The payout layer pays clean stores on the standard schedule and slows, caps, or holds money as a store’s tier climbs.

```

PSEUDOCODE
def payout_treatment(store):
    if store.tier == "disqualified":
        return Action(pay=False, recover_from=store.reserve)
    if store.tier == "probation":

```

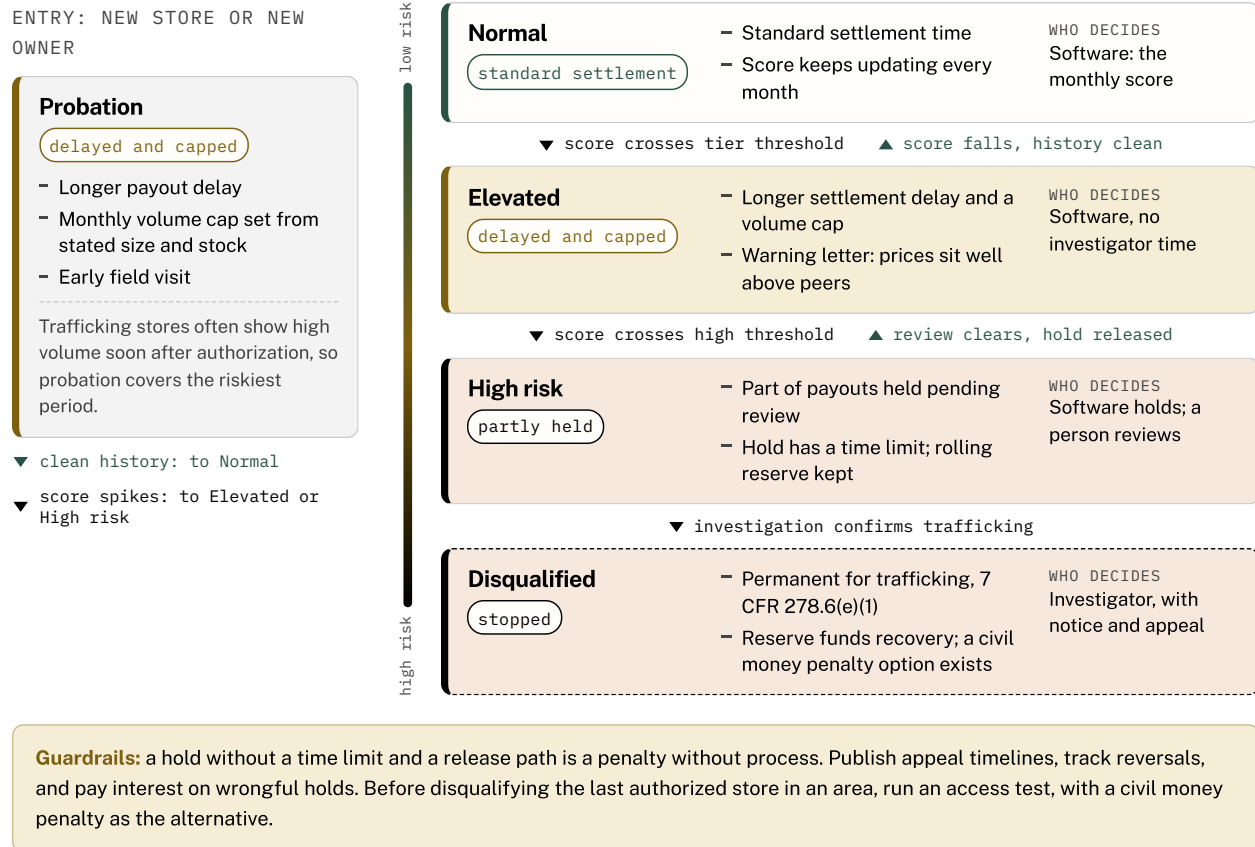


Figure 11: The merchant risk ladder. Like a card processor, the program pays clean stores fast and slows, caps, or holds money as a store's risk score climbs. **Read it as:** software moves a store up and down the first three rungs with no investigator time. Only a person can disqualify, and a week or two of delay leaves money to recover when a case opens.

```

    return Action(delay=PROBATION_DELAY, field_visit="early",
                  monthly_cap=cap_from(store.stated_size,
                                       store.stated_stock))

if store.tier == "normal":
    return Action(delay=STANDARD_SETTLEMENT)
if store.tier == "elevated": # a delay of a week or two
    return Action(delay=ELEVATED_DELAY, monthly_cap=ELEVATED_CAP)
if store.tier == "high_risk":
    return Action(delay=HIGH_RISK_DELAY, monthly_cap=HIGH_RISK_CAP,
                  hold_share=HOLD_SHARE, # hold until review
                  # the hold needs a release path
                  hold_expires=HOLD_TIME_LIMIT,
                  reserve_share=RESERVE_SHARE)

def next_tier(store, score, today):
    # a person decides
    if store.case and store.case.decision == "disqualify":
        return "disqualified"
    if today - store.authorized_or_new_owner_on < PROBATION_PERIOD:
        # trafficking stores spike soon after authorization
        return "probation"

```

```

if score >= HIGH_RISK_THRESHOLD:
    return "high_risk"
return "elevated" if score >= ELEVATED_THRESHOLD else "normal"

```

7.5 Analytics layer

The analytics layer reads from all four transaction-path layers and from third-party sources. It produces store, card, and case scores, and it feeds the investigation queue. Two methods matter most beyond price anomaly detection, which Section 8 covers.

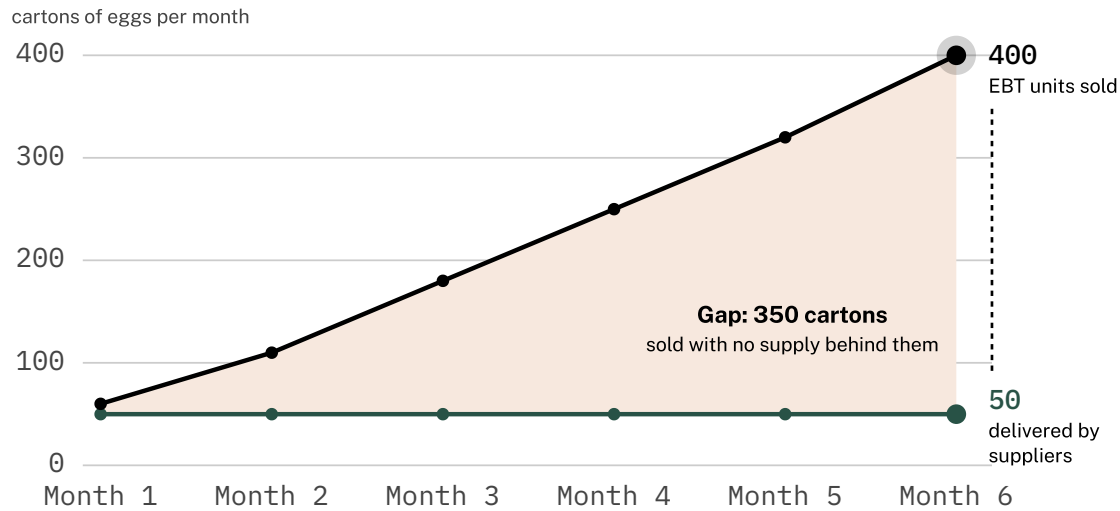
Inventory reconciliation. Compare EBT sales per product category against the store’s wholesale purchases of that category. A store cannot sell 400 cartons of eggs in a month if its suppliers delivered 50. Purchase data can come from supplier invoices that the store submits and from data feeds provided by large distributors under agreement. Without item data, reconciliation works at the level of total food purchases against total EBT redemptions. With item data, it works per product. The limit is that a store can buy stock for cash at a wholesale club, which leaves no supplier record in a distributor feed. The program should make purchase records a condition of authorization: a store must keep invoices or receipts for food it sells and submit them on request. Random audits of a sample of stores each year, plus targeted audits of high scorers, keep the requirement credible. A store that cannot show purchases to support its sales has a problem regardless of what its baskets say. Reconciliation is also the control that survives an adaptive adversary: in the simulation in Section 9, it keeps capture near 60 percent against disguises that price features alone barely detect, as long as purchase records are honest. When half of traffickers forge their invoices, capture falls to between 37.1 and 46.7 percent (Table 9.8), so the invoices themselves need audits (Figure 12).

Reconciliation compares what a store sold through EBT with what its suppliers delivered, evidence the store does not write.

```

PSEUDOCODE
def reconcile_inventory(store, month):
    # Wholesale-club cash buys leave no distributor feed record.
    if not store.purchase_records_submitted(month):
        # a condition of authorization
        request_purchase_records(store, month)
    invoices = (distributor_feeds.invoices(store, month)
               + store.submitted_invoices(month))
    if not store.has_item_data(month):
        sold = sum(t.amount for t in store.ebt_transactions(month))
        bought = sum(i.food_total for i in invoices)
        if sold - bought > TOTAL_TOLERANCE:
            return [Flag(store, month, "ALL_FOOD")]
        return []
    delivered = units_by_upc(line for i in invoices
                             for line in i.invoice_lines)
    sold = units_by_upc(line for t in store.ebt_transactions(month)
                       for line in t.basket_lines)
    flags = []
    for upc, units_sold in sold.items():
        gap = units_sold - delivered.get(upc, 0)
        if gap > UNIT_TOLERANCE:
            flags.append(Flag(store, month, upc, gap))
    return flags

```



Illustrative Month 6 matches the report's example: 400 sold, 50 delivered. Months 1 to 5 are invented to show the trend.

The check

EBT sales per product category against the store's wholesale purchases of that category, from supplier invoices and distributor feeds.

Caveat: cash buys

A store can buy stock for cash at a wholesale club. That leaves no record in a distributor feed. So authorization requires:

- 1 purchase records, kept and submitted on request
- 2 random audits of a sample of stores each year
- 3 targeted audits of high scorers

Figure 12: Sales need supply. EBT sales of a product cannot outrun what suppliers delivered, so a growing gap points to fake or inflated baskets. **Read it as:** the red band is EBT sales with no recorded supply behind them. A store that cannot show purchases to support its sales has a problem, whatever its baskets look like.

Report example: 400 cartons of eggs sold, 50 delivered.

Network and graph analysis. Build a graph with cards and stores as nodes and transactions as edges. Trafficking rings leave clusters: groups of cards that all visit the same few suspicious stores, often traveling past closer and larger stores to do so. A single-transaction rule sees a normal-looking \$60 sale. The graph sees 200 cards that spend nearly all their benefits at three small stores owned by related parties. Community detection, shared-owner links from authorization records, and shared

bank accounts on payout records together reveal rings that no store-level score would. The same graph helps on the recipient side: a card that appears in many confirmed trafficking cases is a candidate for a state investigation with evidence beyond a single transaction pattern (Figure 13).

The ring finder builds a card to store graph and flags groups of cards whose spending piles into the same few stores.

```
PSEUDOCODE
def find_rings(transactions):
    graph = Graph()
    for txn in transactions:
        graph.add_edge(txn.card, txn.store, weight=txn.amount)
    rings = []
    for cluster in community_detection(graph):
        stores = cluster.stores_by_spend()[:FEW_STORES]
        cards = [c for c in cluster.cards
                  if graph.spend_share(c, stores) >= CONCENTRATION_SHARE]
        if len(cards) < MIN_RING_CARDS:
            continue
        # Rings travel past closer and larger stores.
        skipped = count_cards_passing_closer_larger_stores(cards, stores)
        linked = shared_owners(stores) or shared_payout_accounts(stores)
        rings.append(Ring(stores, cards, skipped, linked))
    return rings

def recipient_leads(graph, confirmed_cases):
    # Evidence beyond one transaction pattern, for state investigators.
    return [c for c in graph.cards
            if confirmed_cases.count_with(c) >= RECIPIENT_CASE_COUNT]

# Report example: 200 cards, nearly all benefits at three stores.
```

Feedback. Every closed case, whether confirmed or cleared, becomes a labeled example. The scoring models retrain on these labels. Cleared cases matter as much as confirmed ones, because they teach the models which patterns are benign, such as round prices at a store that prices in whole dollars.

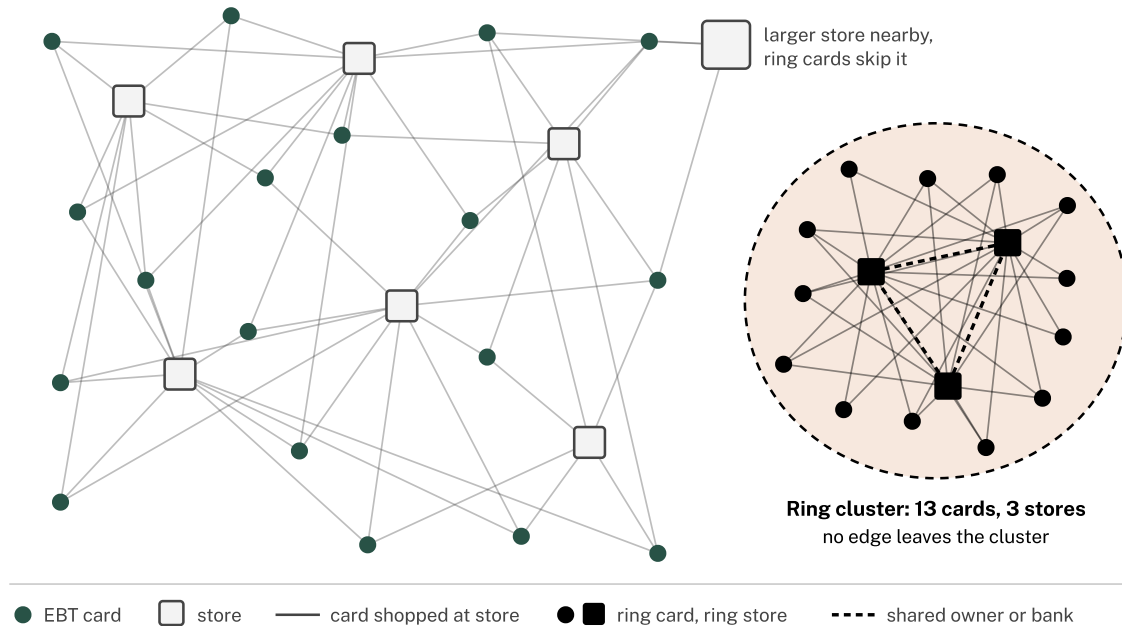
8 Price anomaly detection without human basket review

Item-level data creates a new problem: volume. Retailers process millions of SNAP transactions per day. No program can afford a person to look at baskets. The design question is how to turn that volume into a short list of stores worth a human’s time, so that operating cost grows with investigator headcount while transaction volume affects only computing cost. Figure A8 in Appendix A shows the monthly scoring run as a sequence.

8.1 Reference prices

The system builds a reference price for each barcode in each region from all stores’ basket data. For a given barcode and region in a given month, the reference is a robust central value of observed prices, such as the median, with a spread measure such as the interquartile range. Separate references for store classes (supermarket, convenience store, small grocery) reflect the fact that convenience stores charge more for the same item. Products with few observations borrow strength from related products in the same category and from neighboring regions (Figure 14).

Normal shopping: each card spreads across many stores



Single-transaction rule

Sees one \$60 sale at a licensed store.
Normal amount, normal store. It passes.

Graph view

Report example: 200 cards spend nearly all
their benefits at three small stores owned
by related parties, often passing closer and
larger stores.

Signals that expose a ring

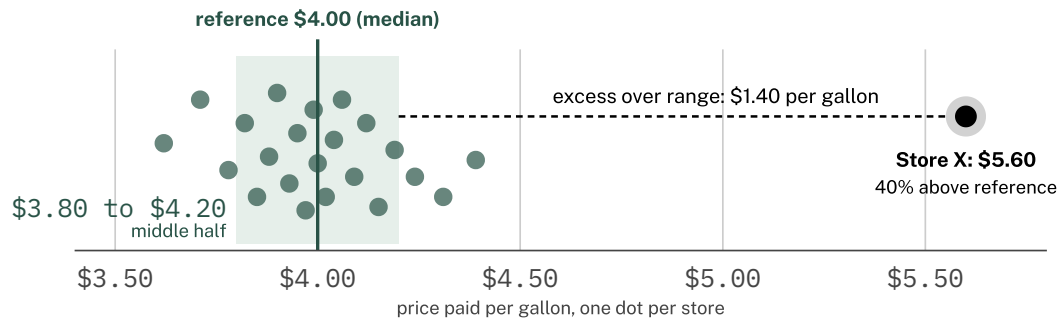
community detection shared owners
shared bank accounts

Owners come from authorization records,
bank accounts from payout records.

Figure 13: The graph sees what one sale hides. With cards and stores as nodes and sales as edges, a ring shows up as a closed cluster of cards around a few stores. **Read it as:** each ring sale looks normal alone. The evidence is the shape: many cards that shop only at the same three linked stores. The graph shows a sample; the report's example ring has 200 cards.

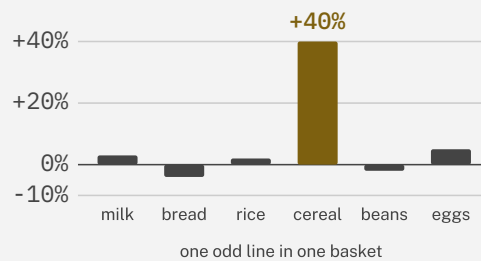
A. ONE BARCODE, ONE REGION A gallon of milk, one month, one store class

Illustrative



B1. NOISE

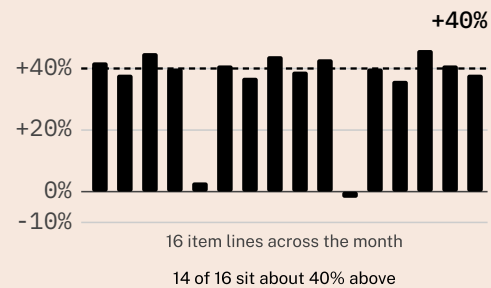
One odd basket means little



A promotion, a slow item priced high, or a mis-keyed price. Scoring each basket floods the system with noise.

B2. SIGNAL

A store-month pattern



Most of the store's lines sit about 40% above reference for a whole month. Shrinkage keeps small stores from extreme scores.

Bars show price vs reference, where 0 is the reference price. The reference uses the median and the middle half of all stores' prices, set per store class; rare products borrow from related products and neighboring regions.

Figure 14: Score the store, not the basket. Each barcode gets a regional reference price, and a store whose lines sit far above it month after month is the signal. **Read it as:** Store X's single \$5.60 price is one dot. What matters is panel B2: the same store high on most lines all month.

8.2 Score stores, not baskets

One odd price means little. A store may run a promotion, price a slow item high, or key a price wrong. Independent stores set their own prices, and small stores pay more at wholesale than chains. Scoring each basket would flood the system with noise.

The unit of scoring is the store-month. For each store and month, the system computes:

- The excess price share: the sum over lines of (price paid minus the upper bound of the reference range, when positive), divided by total EBT sales.
- The share of lines priced above the reference range.
- Basket plausibility: how far the mix of items departs from what EBT shoppers buy at similar stores (for example, baskets made of a few high-priced items that recur in round-total combinations).
- Supply consistency: EBT sales per category divided by known purchases per category, from the inventory reconciliation.

These features feed a model trained on labels from past cases. Before scoring, the system applies statistical shrinkage so that a store with few transactions does not get an extreme score from a handful of odd lines. The simulation in Section 9 finds no measurable effect from this step in its setting, so it remains a cheap safeguard whose value is untested.

The monthly run builds a reference price per barcode and region, then scores each store-month from all of its lines together.

PSEUDOCODE

```
def reference_prices(lines, month):
    refs = {}
    groups = group(lines, by=(upc, region, store_class))
    for (upc, region, store_class), prices in groups:
        if len(prices) < MIN_OBSERVATIONS:
            # borrow from category and neighbors
            prices += related_prices(category_of(upc),
                                    neighboring_regions(region))
        mid, spread = median(prices), interquartile_range(prices)
        upper = mid + REF_RANGE_WIDTH * spread
        refs[(upc, region, store_class)] = Reference(mid, upper=upper)
    return refs

def score_store_month(store, month, refs, reconciliation, model):
    lines = store.ebt_basket_lines(month)
    upper = {l: refs[(l.upc, store.region, store.store_class)].upper
             for l in lines}
    sales = sum(l.quantity * l.unit_price for l in lines)
    excess = sum(max(0, l.unit_price - upper[l]) * l.quantity
                 for l in lines)
    above = count(l for l in lines if l.unit_price > upper[l])
    features = {
        "excess_price_share": excess / sales,
        "share_lines_above": above / len(lines),
        "basket_plausibility": mix_distance(store.baskets(month),
                                             peers(store.store_class)),
        "supply_consistency":
            reconciliation.sales_vs_purchases_by_category(store, month),
    }
    raw = model.predict(features) # trained on labels from past cases
    # Shrinkage: meant to stop one odd basket swinging a small store
```

```
# (untested, see Section 9).
score = shrink(raw, toward=peer_average, n=len(lines),
               strength=SHRINKAGE_STRENGTH)
return StoreMonthScore(store, month, score, features)
```

8.3 Rank by expected loss

The output of scoring is a ranked list of stores by expected loss:

$$E = p \times S \times e \times 12$$

where

- E a store's expected loss per year
- p its probability of fraud
- S its monthly EBT sales
- e its excess share

The probability p comes from the model. The excess share e comes from the excess price share and the supply gap, and $S \times e \times 12$ is the store's estimated excess dollars per year. The case for expected loss is that stores differ in size, so the same probability puts very different sums at risk. The simulation in Section 9 finds no difference between this ranking and ranking by probability alone in its setting, so the case rests on that argument, not on evidence. The program then acts on the list through a funnel ([Figure 15](#)).

The automated tier handles most stores that score above normal. A warning letter tells a store its prices sit well above its peers and invites an explanation. A payout delay or volume cap limits exposure without accusing anyone. These actions are reversible and need no investigator time. They also deter: a store that knows the system watches its prices has less reason to inflate them.

8.4 Cost model

Human cost in this design scales with the number of investigators. Transaction volume drives only computing cost, which is small. That property follows from three rules.

First, set the investigator budget first. The program decides how many investigator hours it can fund in a period. That number fixes N , the count of cases the queue can absorb.

Second, review only what the budget covers. The queue takes the top N stores by expected loss. Stores below the cutoff stay in the automated tier, where their scores keep updating. A store that keeps rising will eventually enter the queue.

Third, review a case only when the expected benefit exceeds the review cost. Even within budget, a case with low expected loss should not consume a review. The test is:

$$E \times r > C$$

where

- r the share of the expected loss that a review prevents or recovers
- C the cost of one review

The share r reflects that a review does not recover every dollar. It may end future losses through disqualification, recover part of past losses, and deter others.

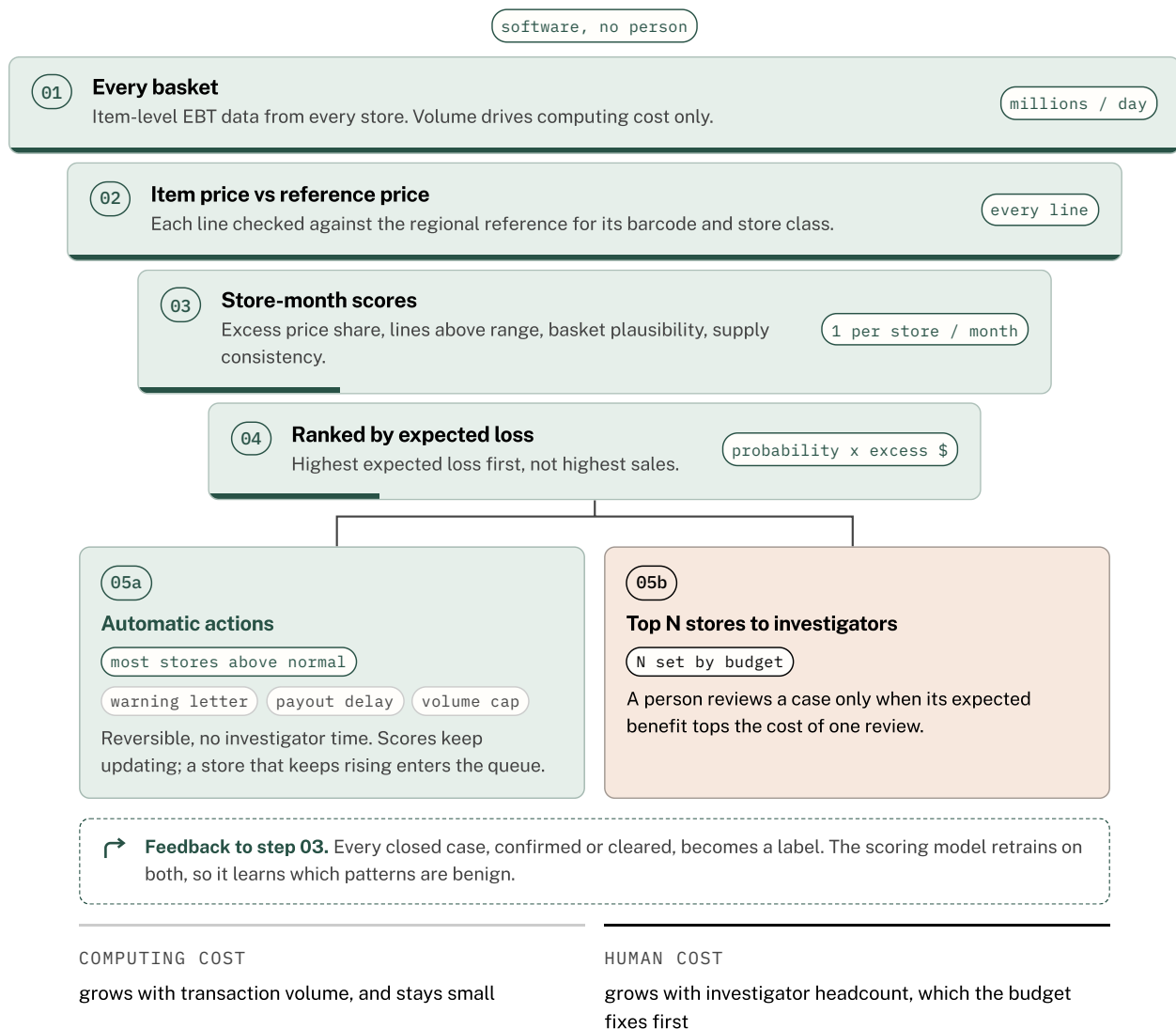


Figure 15: The review funnel. Software shrinks millions of daily baskets to a short list, and people see only the top N stores the budget can cover. **Read it as:** every step above the fork runs in software, so more baskets cost only computing. People enter at one point, the top N, and the budget sets N.

CAUTION

Worked example (illustrative numbers only). These figures are invented to show the method. They are not estimates of real store behavior or real investigation costs. Assume a review costs \$6,000 in investigator time (C), and a confirmed case prevents or recovers half of the expected annual loss ($r = 0.5$).

Table 8.1. Worked example of expected loss and the review test (illustrative numbers)

Store	Monthly EBT sales	Estimated excess share	Excess dollars per year	Probability of fraud	Expected loss per year	Expected benefit of review (50%)	Review?
A	\$60,000	30%	\$216,000	0.8	\$172,800	\$86,400	Yes, rank 1
B	\$25,000	40%	\$120,000	0.6	\$72,000	\$36,000	Yes, rank 2
C	\$120,000	5%	\$72,000	0.3	\$21,600	\$10,800	Passes threshold; budget decides
D	\$15,000	20%	\$36,000	0.4	\$14,400	\$7,200	Passes threshold; budget decides
E	\$8,000	10%	\$9,600	0.2	\$1,920	\$960	No; automated tier only

For Store A, the expected loss of Section 8.3 and the review test give:

$$E = 0.8 \times \$60,000 \times 0.30 \times 12 = \$172,800$$

$$E \times r = \$86,400 > C = \$6,000$$

Store E fails the threshold: a \$6,000 review to protect \$960 is a loss. It receives a warning letter and stays under watch. Stores A through D all pass. If the budget covers two reviews this cycle, A and B go to investigators. C and D receive payout delays and volume caps, and they remain at the top of next month's list. Note that Store C has the largest sales but a low excess share and a low probability. Ranking by sales volume alone, a common heuristic, would have sent the wrong store first. The simulation in Section 9 supports that point: ranking by sales volume caught none of the trafficked dollars. Ranking by probability alone would swap C and D in this example, but in the simulation that kind of difference did not change how much the queue caught (Figure 16).

The planner ranks stores by expected loss, sends the budget's top N to people, and puts the rest on automatic actions.

PSEUDOCODE

```
def plan_reviews(scores, investigator_hours):
    # budget fixes N before the run
    n = investigator_hours // HOURS_PER_REVIEW
    for s in scores:
        # Excess share comes from excess prices and supply gap.
        s.expected_loss = (s.probability_of_fraud * s.monthly_ebt_sales
                          * s.excess_share * 12)
    ranked = sorted(scores, key=expected_loss, descending=True)
```

```

worth = [s for s in ranked
         if s.expected_loss * SHARE_PREVENTED > REVIEW_COST]
for s in worth[:n]:
    # back on next list
    apply(s.store, [Action("payout_delay"), Action("volume_cap")])
for s in ranked:
    if s not in worth and s.score > NORMAL_SCORE:
        apply(s.store, [Action("warning_letter")])
return [Case(s.store, s.expected_loss) for s in worth[:n]]

def close_case(case):
    # Cleared cases teach the model which patterns are benign.
    training_labels.add(case.store_month_score, label=case.outcome)

# Worked example: REVIEW_COST 6,000, SHARE_PREVENTED 0.5, N = 2
# loss = p x sales x share x 12; worth a review if loss x 0.5 > 6,000
#      p   sales  share   loss   x 0.5  action
# A: 0.8   60,000  0.30  172,800  86,400  review, rank 1
# B: 0.6   25,000  0.40   72,000  36,000  review, rank 2
# C: 0.3  120,000  0.05   21,600  10,800  delay and cap, past N
# D: 0.4   15,000  0.20   14,400   7,200  delay and cap, past N
# E: 0.2    8,000  0.10    1,920    960  warning letter

```

8.5 Recipient-side reporting

Recipients can supply a free signal against overcharging. The recipient app should show an itemized receipt for each EBT purchase, with a “price looks wrong” button on each line. Reports feed the store’s score as one more feature, weighted by the reporter’s history so that one angry customer cannot sink a store. Reports also help the household directly, by showing what each item cost.

This channel does nothing against trafficking. The trafficking shopper colludes with the store and will never press the button. Recipient reporting is a tool for the predatory form of overcharging, where the store cheats its customer. That distinction should shape how the program weights these reports: they are strong evidence of predatory pricing and no evidence about trafficking.

9 Evaluation

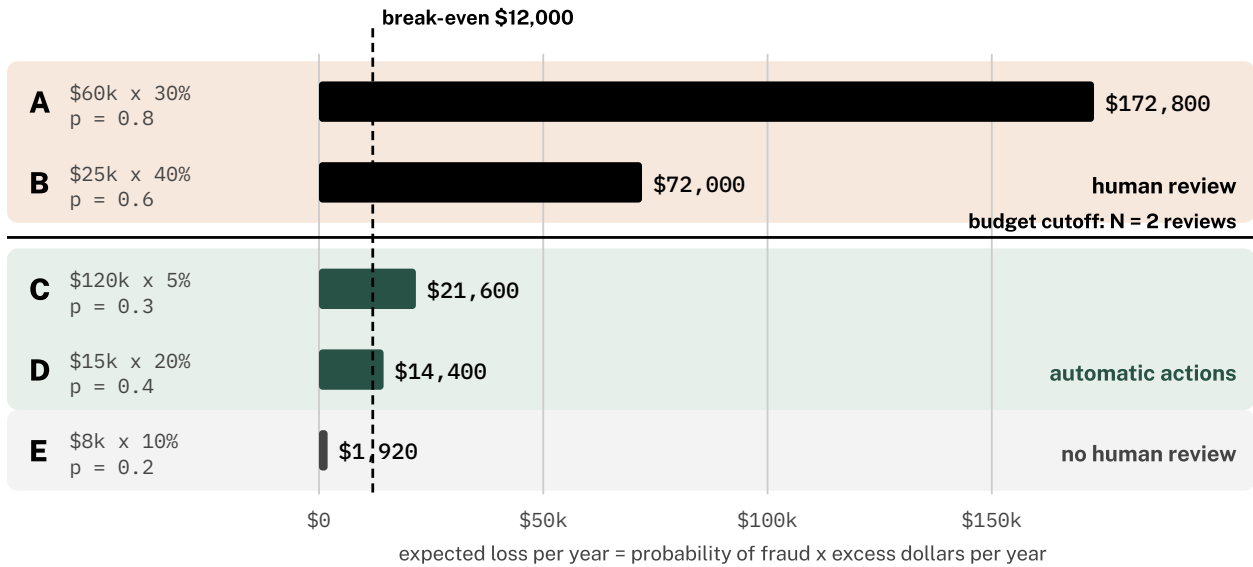
No real data can test the method in Section 8, because EBT records today carry no item lines. This section therefore tests it in a synthetic world. Every number in this section is simulated. None estimates real trafficking, detection, costs, or error rates. The results show how the method behaves under the stated assumptions.

9.1 Method

The world. The simulation builds two neighboring price regions with 350 stores in three classes: supermarkets, small groceries, and convenience stores. Stores sell from a 300-barcode catalog, and the world runs for six months of item-level EBT baskets. Honest pricing carries the noise Section 8.2 describes: class markups, promotions, high prices on slow items, mis-keyed prices, small stores that pay more at wholesale, and small stores that price in whole dollars. Some purchases leave no supplier record, which stands in for cash buys at a wholesale club. Only small stores traffic, in line with the FNS finding that small stores carry over 95 percent of trafficked dollars [50]. A trafficker either keys fake baskets for cash swipes or overcharges real baskets.

Illustrative, Section 7.3

Review if $\text{expected loss} \times 50\% \text{ recovered} > \$6,000$ per review, so expected loss must top **\$12,000**.



Human review: A, B

Above break-even and inside a budget of two reviews.

Automatic actions: C, D

Pass the threshold, fall outside budget. Payout delays and volume caps; they top next month's list.

No human review: E

A \$6,000 review to protect \$960 is a loss. Warning letter, stays under watch.

Sales volume misleads

Store C has the largest sales, \$120k a month, but ranks third. Ranking by sales would send the wrong store first.

Figure 16: Rank stores by expected loss. Review cost sets the break-even line, and the budget sets how many stores above it get a person. **Read it as:** bars left of the dashed line cost more to review than they could recover. Of the stores to its right, only the top N fit the budget; the rest get automatic, reversible actions.

Calibration. One factor scales every trafficker’s intensity so that trafficked dollars equal 1.8 percent of benefit dollars. That is the midpoint of the published range of 1.6 to 2.0 percent [50]. The realized rate is 1.8 percent in all 20 seeds. Small stores redeem 15.3 percent of EBT dollars, close to the 15 percent in the same study. A world holds 46.6 trafficking stores on average. The 1.8 percent rate is an input, not a finding.

The pipeline. The code implements Sections 8.1 to 8.4 month by month. It builds reference prices per barcode, region, and store class from the median and interquartile range. A sparse cell borrows from its category and then from the neighboring region. The code computes the four store-month features of Section 8.2. A logistic regression gives the probability of fraud. It trains only on cases closed in earlier months: the stores reviewed in those months plus six random audits a month. Shrinkage pulls the probability toward the class peer average, weighted by line count. Expected loss is probability times monthly sales times excess share times 12. Excess share is the larger of the excess price share and the supply gap. The planner applies the threshold test with the illustrative values of the worked example in Section 8.4: a review cost of \$6,000 and a share prevented of 50 percent. It sends the top N to review. Stores past N get a payout delay and a volume cap. Stores below the threshold with a probability above 0.5 get a warning letter.

Idealized and realistic runs. Tables 9.1 to 9.5 describe an idealized program. Every review finds the truth, six random audits a month supply unbiased labels, supplier invoices are honest, and all traffickers share one disguise. Tables 9.6 to 9.9 relax each of these assumptions and then combine them in one realistic scenario.

Metrics. Evaluation runs on month 6, after five months of closed cases. The main metric is the share of trafficked dollars held by the reviewed stores. The others are precision (the share of reviewed stores that traffic), recall (the share of trafficking stores reviewed), dollars prevented, and the share of honest stores that receive each action. Each cell shows the mean over 20 seeds and a 90 percent interval, the 5th to 95th percentile across seeds.

Reproduce. The evaluation needs only Python 3 and its standard library: [download the code](#)¹, whose README lists every parameter with its source, unzip it, and run:

```
SHELL
# rebuild every table from fixed seeds
python3 run.py
# rerun; fail on any changed byte
python3 run.py --check
```

9.2 Results

Unless stated otherwise, results use the crudest trafficker (few lines, large markups, round totals) and a budget of $N = 15$.

Table 9.1 shows how detection and dollars prevented grow with the review budget. Dollars prevented are annualized and use the true trafficked dollars of the reviewed traffickers. Net benefit subtracts the review cost.

Table 9.1. Detection and cost against budget

¹<https://truonghan.com/designing-out-snap-fraud/designing-out-snap-fraud-eval.zip>

N	Stores reviewed	Share of trafficked \$ caught	Precision	Recall	True \$ prevented per year	Net benefit
1	1.0	9.5% 5.7 to 19.0	100.0%	2.2%	\$202k	\$196k
5	5.0	34.4% 28.1 to 43.9	100.0%	11.0%	\$725k	\$695k
10	10.0	53.1% 40.4 to 60.6	100.0%	21.9%	\$1,118k	\$1,058k
15	15.0	67.3% 55.2 to 73.8	99.7%	32.8%	\$1,413k	\$1,323k
20	20.0	76.8% 68.3 to 86.1	99.2%	43.6%	\$1,613k	\$1,493k
25	24.7	84.1% 75.1 to 90.1	98.6%	53.3%	\$1,765k	\$1,616k
30	28.6	88.2% 79.4 to 92.9	96.8%	60.1%	\$1,852k	\$1,681k
40 to 60	30.9	90.2% 85.8 to 93.2	95.6%	63.7%	\$1,896k	\$1,711k

At $N = 15$, review reaches stores holding 67.3 percent of trafficked dollars at 99.7 percent precision, but only 32.8 percent of trafficking stores. The queue reaches the large traffickers first. The marginal benefit of one more review falls from \$152.0k at $N = 2$ to \$18.6k at $N = 40$, and it never drops below the \$6,000 review cost. From $N = 30$, the budget no longer ends the queue. The threshold test does. Too few stores pass it, so the mean number of reviews falls short of N (28.6 at $N = 30$) and stops at 30.9 from $N = 40$, with recall flat at 63.7 percent.

Table 9.2 compares the essay’s ranking with three alternatives at $N = 15$. The baselines take the top N on their own key, without the threshold test.

Table 9.2. Ranking baselines at $N = 15$

Ranking	Share of trafficked \$ caught	Precision	Recall	True \$ prevented per year
Expected loss (Section 8)	67.3% 55.2 to 73.8	99.7%	32.8%	\$1,413k
Probability only (shrunk)	66.6% 56.4 to 73.5	99.0%	32.6%	\$1,399k
EBT sales volume	0.0% 0.0 to 0.0	0.0%	0.0%	\$0k
Random	2.9% 0.0 to 7.7	12.0%	3.6%	\$63k

Ranking by EBT sales volume catches nothing. Supermarkets top that list, and none of them traffic in this world. This is the Store C point of Section 8.4 at scale. Random review catches 2.9 percent of trafficked dollars. Ranking by probability alone catches 66.6 percent, within noise of expected loss at 67.3 percent.

Table 9.3 removes one piece at a time at $N = 15$. The adaptive adversary keys fake baskets with many lines at shelf prices and non-round totals, and it overcharges by billing extra units, not higher prices.

Table 9.3. Ablations at $N = 15$

Variant	Share of trafficked \$ caught	Precision	Recall	Honest stores reviewed
Method as in Section 8	67.3% 55.2 to 73.8	99.7%	32.8%	0.1
Shrinkage off	66.8% 55.0 to 73.4	100.0%	32.9%	0.0
Reconciliation off	67.3% 56.5 to 73.5	97.7%	32.2%	0.3
Adaptive adversary	57.8% 42.8 to 70.9	93.3%	30.9%	1.0
Adaptive adversary, reconciliation off	0.0% 0.0 to 0.0	0.0%	0.0%	2.5

Turning shrinkage off changes nothing measurable. Turning reconciliation off also changes little against the crude trafficker, whose prices give it away. The adaptive adversary lowers capture to 57.8 percent, because reconciliation still sees sales the store never bought. With reconciliation also off, capture falls to zero.

Table 9.4 is the main idealized result. It keeps the same stores and the same calibrated trafficked dollars and changes only the disguise. It runs each disguise with supply reconciliation on and off.

Table 9.4. Detection against trafficker disguise

Disguise	Reconciliation	Share of trafficked \$ caught	Precision	Honest stores actioned
Gross: few lines, huge markup, round total	on	67.3% 55.2 to 73.8	99.7%	0.8%
Gross: few lines, huge markup, round total	off	67.3% 56.5 to 73.5	97.7%	0.9%
Moderate: 1.2 to 1.5x markup over many lines	on	61.2% 46.8 to 72.0	95.0%	4.2%
Moderate: 1.2 to 1.5x markup over many lines	off	19.2% 0.0 to 36.0	55.5%	8.4%
Edge: just above last month's reference range	on	62.5% 49.8 to 72.1	96.3%	4.0%
Edge: just above last month's reference range	off	14.4% 0.0 to 33.9	42.0%	8.0%
Quantity: extra units at shelf price	on	57.8% 42.8 to 70.9	93.3%	5.1%
Quantity: extra units at shelf price	off	0.0% 0.0 to 0.0	0.0%	11.1%

With price features alone, capture falls from 67.3 percent against the crude trafficker to between 0 and 19.2 percent against the three subtler disguises. With supply reconciliation on, capture stays between 57.8 and 62.5 percent. Price features catch only the crude disguise. Reconciliation does the work against anything subtler. Subtle disguises also raise the load on honest stores. The share of honest stores given any action rises from under 1 percent against the crude trafficker to 4.0 to 5.1 percent with reconciliation on, and 8.0 to 11.1 percent with it off.

Table 9.5 shows the share of honest stores in each group that receive each action at $N = 15$. High wholesale cost marks honest small stores whose prices sit about 15 percent above their class.

Table 9.5. False-positive load on honest stores at $N = 15$

Honest group	Human review	Payout delay or cap	Warning letter	Any action
Supermarket	0.2%	2.7% 0.0 to 10.0	0.0%	2.8%
Small grocery	0.0%	0.4%	0.4%	0.8%
Convenience	0.0%	0.1%	0.2%	0.3%
Small, high wholesale cost	0.0%	0.4%	0.9%	1.2% 0.0 to 4.3
Small, normal cost	0.0%	0.2%	0.2%	0.4% 0.0 to 1.5

No honest small store reaches human review at $N = 15$ in any seed. Honest small stores with high wholesale costs receive some automated action 1.2 percent of the time, against 0.4 percent for small stores with normal costs. Honest supermarkets receive a payout delay or cap 2.7 percent of the time, and 10.0 percent at the 95th percentile. The cause is the formula: expected loss scales with sales, so a modest score at a very large store can pass the threshold test. These false flags are why the payout layer in Section 7.4 needs time limits and a clear path to release.

The remaining tables drop the idealized assumptions. In them, a trafficked dollar counts as caught only when the review of its store substantiates the case, and net benefit uses only those dollars.

Table 9.6 makes investigations imperfect. A review substantiates a true trafficker with probability d and wrongly substantiates an honest store with probability f . The model retrains each month on these findings, not on the truth, and the evaluation month’s reviews carry the same errors. The quantity columns use the adaptive adversary of Table 9.3.

Table 9.6. Imperfect investigations at $N = 15$

Detect (d)	False substantiation (f)	Caught, crude	Caught, quantity	Precision, quantity	Honest stores wrongly substantiated, quantity	Net benefit, quantity
1.0	0.00	67.3%	57.8% 42.8 to 70.9	93.3%	0.0	\$1,127k
1.0	0.01	67.0%	57.6% 42.6 to 70.9	92.7%	0.3	\$1,122k
1.0	0.03	66.5%	57.6% 42.7 to 70.8	92.3%	1.2	\$1,122k
0.8	0.00	49.9%	43.7% 30.5 to 61.2	89.7%	0.0	\$830k
0.8	0.01	49.8%	43.1% 24.0 to 61.2	87.7%	0.2	\$816k
0.8	0.03	47.7%	42.2% 23.6 to 59.6	84.7%	1.2	\$800k
0.6	0.00	34.6%	32.6% 20.5 to 43.9	87.7%	0.0	\$599k
0.6	0.01	33.6%	31.6% 17.3 to 43.9	84.3%	0.3	\$576k
0.6	0.03	32.5%	31.3% 19.0 to 43.3	81.7%	1.2	\$570k

Missed traffickers cost the most. At $d = 0.8$, capture falls from 67.3 to 49.9 percent against the crude trafficker and from 57.8 to 43.7 percent against the quantity disguise. At $d = 0.6$ it falls to 34.6 and 32.6 percent. Part of the loss is direct, because a missed case prevents nothing. Part is learning. A cleared trafficker becomes a negative label, so the queue itself gets worse: against the crude trafficker, precision falls from 99.7 to 89.3 percent at $d = 0.8$. False substantiation barely moves capture. Its cost falls on honest stores. At $f = 0.03$, about 1.2 honest stores per world are wrongly found to traffic over six months. Net benefit stays positive in every cell, at the 5th percentile too.

Table 9.7 asks what happens when labels exist only for stores someone investigated. The queue picks likely traffickers, so its labels overstate how common trafficking is. Random audits pick stores at random and give unbiased labels. They sit on top of the 12 queued reviews a month. The oracle trains on every store’s true label, a bound no program reaches. Reviews are perfect here, to isolate the effect.

Table 9.7. Label sources and random audits at $N = 15$

Labels	Caught, crude	Caught, quantity	Precision, quantity	Honest stores actioned, quantity
Reviewed stores only	66.4%	58.2%	90.7%	30.3% 9.3 to 52.3
Plus 1 random audit a month	67.1%	58.3%	92.3%	13.2% 4.2 to 26.5
Plus 6 random audits a month (Tables 9.1 to 9.5)	67.3%	57.8%	93.3%	5.1% 0.7 to 11.5
Oracle: every store’s true label	67.1%	57.2%	93.3%	1.3% 0.3 to 2.8

Selection bias does not hurt capture. Labels from reviewed stores alone catch 66.4 percent against the crude trafficker and 58.2 percent against the quantity disguise, within noise of the other arms. It hurts honest stores. A model trained only on reviewed stores, most of which traffic, overstates every store’s probability of fraud. Against the quantity disguise, 30.3 percent of honest stores then get a warning letter, a payout hold, or a review. One random audit a month cuts that to 13.2 percent, six cut it to 5.1 percent, and the oracle reaches 1.3 percent. Random audits help, but not by finding more trafficking. They keep the probabilities calibrated, and the warning threshold depends on those probabilities.

Table 9.8 lets a share of traffickers forge supplier invoices. A forger’s recorded purchases grow by exactly its trafficked dollars, so reconciliation sees no gap. Reconciliation is on and reviews are perfect. The disguises are the three subtle ones of Table 9.4, where reconciliation carries the result.

Table 9.8. Forged supplier invoices at $N = 15$

Disguise	Caught, no forgers	Caught, 25% forge	Caught, 50% forge	Precision, 50% forge
Moderate: 1.2 to 1.5x markup over many lines	61.2%	53.2%	44.4% 28.9 to 58.0	78.2%
Edge: just above last month’s reference range	62.5%	56.4%	46.7% 37.2 to 59.9	80.6%
Quantity: extra units at shelf price	57.8%	48.6%	37.1% 23.8 to 48.3	71.8%

Capture falls roughly in step with the share of forgers. With half of traffickers forging, it falls to 44.4 percent (moderate), 46.7 percent (edge), and 37.1 percent (quantity). A forger looks like a store with reconciliation off, and Table 9.4 shows how little the price features catch then. Reconciliation is only as strong as the purchase records behind it. Distributor feeds, which the store does not write, are worth more than invoices the store submits.

Table 9.9 stacks the departures one at a time, from the idealized run to a realistic one. The adversary mix gives each trafficker one of the four disguises of Table 9.4, with equal odds. The realistic row uses $d = 0.8$, $f = 0.01$, labels from reviewed stores only, and forged invoices at 25 percent of traffickers. These values are assumptions, not measurements.

Table 9.9. From the idealized run to a realistic scenario at $N = 15$

Setting	Share of traf- ficked \$ in re- viewed stores	Share of trafficked \$ caught	Precision	Honest stores wrongly substanti- ated	Honest stores actioned	Net benefit
Idealized (Tables 9.1 to 9.5)	67.3%	67.3% 55.2 to 73.8	99.7%	0.0	0.8%	\$1,323k
Adversary mix	62.1%	62.1% 51.5 to 70.6	95.3%	0.0	3.7%	\$1,221k
Add imperfect reviews	56.3%	44.9% 32.4 to 64.6	85.0%	0.3	2.3%	\$856k
Add labels from reviewed stores only	54.8%	43.2% 25.1 to 62.0	78.7%	0.3	4.7%	\$824k
Add 25% forgers: realistic	47.9%	38.6% 24.5 to 58.5	75.7%	0.2	5.2%	\$725k \$372k to \$1,225k
Realistic, plus 6 random audits a month	48.7%	39.3% 23.6 to 61.6	79.2%	0.4	2.6%	\$738k
Realistic, reconciliation off	14.9%	13.1% 1.8 to 24.9	73.3%	0.1	5.7%	\$241k

The realistic scenario weakens the result. Capture falls from 67.3 to 38.6 percent, precision from 99.7 to 75.7 percent, and net benefit from \$1,323k to \$725k. The share of honest stores given any action rises from 0.8 to 5.2 percent. Imperfect reviews cost the most, taking capture from 62.1 to 44.9 percent. The queue gets worse, from 62.1 to 56.3 percent of trafficked dollars in reviewed stores, and missed cases cost the rest. Three findings survive. Net benefit stays positive, with a 90 percent interval of \$372k to \$1,225k. Reconciliation still carries the result: with it off, the realistic setting catches 13.1 percent. Six random audits a month cut the honest-store action rate from 5.2 to 2.6 percent, while capture stays at 39.3 percent.

9.3 What the simulation does not support

Five results limit what this paper can claim.

First, the headline capture rate rests on idealized assumptions. With perfect reviews and honest invoices, reconciliation keeps capture between 57.8 and 62.5 percent against disguised trafficking (Table 9.4). In the realistic scenario of Table 9.9, capture is 38.6 percent and precision 75.7 percent.

The supported claim is narrower: reconciliation-based review pays for itself and beats price features alone. The claim that review reaches about 60 percent of trafficked dollars holds only in the idealized setting.

Second, ranking by expected loss does not beat ranking by probability alone in this world. The two catch 67.3 and 66.6 percent of trafficked dollars, well inside the spread across seeds. Expected loss should matter most when stores of very different sizes have similar scores. A likely reason it does not matter here is that every trafficker is a small store. Section 8.3 keeps expected loss on the strength of its argument, not on this evidence.

Third, shrinkage shows no measurable effect. Capture is 67.3 percent with it and 66.8 percent without it, and honest stores reviewed are 0.1 with it and 0.0 without it. Its purpose is to stop a small store with a few odd lines from jumping the queue. It stays in the design as a cheap safeguard whose value this study does not show.

Fourth, the crude trafficker is easy to catch by construction. It keys a few lines at grossly inflated prices, so precision near 100 percent in Tables 9.1 to 9.3 says more about that adversary than about the method. Tables 9.4 and 9.9 show what survives subtler disguises and a mix of them. The answer is reconciliation, not the price features.

Fifth, random audits do not raise capture. In Tables 9.7 and 9.9 they protect honest stores by keeping the model's probabilities calibrated. Their case rests on fairness to honest stores, not on catching more trafficking.

9.4 Threats to validity

Tables 9.6 to 9.9 now test four assumptions that Tables 9.1 to 9.5 make: perfect reviews, unbiased labels, honest invoices, and one adversary type. Each test rests on further choices, listed below with the ones still untested.

Internal validity. Seven choices inside the simulation shape its results.

- The 1.8 percent trafficking rate holds by construction. One scaling factor sets every trafficker's intensity to meet it.
- Review errors are independent. Table 9.6 draws each review's error at random. Real errors likely correlate with the store: an investigator may clear stores whose patterns look benign, which are the stores the model already scores low. Correlated errors would teach the model its own blind spots. This study does not test them.
- Random audits sit outside the review budget. Table 9.7 adds them on top of the 12 queued reviews a month. A program that pays for audits from the review budget would trade some capture for them. The oracle is a bound, not a policy.
- Forgery is all or nothing. In Table 9.8 a forger covers exactly its trafficked dollars, and every other trafficker's records stay honest. Partial forgery, or stock bought for cash to back fake sales, would land in between. Reconciliation also works at store level, not per category.
- The excess-share formula is the author's choice. It takes the larger of the excess price share and the supply gap, with the gap measured against the 75th percentile of the sales-to-purchases ratio in the store's class. Section 8.3 says excess dollars come from both but does not say how to combine them.
- The model can start untrained. In the cell with the quantity disguise and reconciliation off, some seeds found no trafficker in the label months. With no positive labels, the code fell back to a cold-start probability of 1 for every store. Every store then sat above the warning threshold of 0.5. That is why the honest-action interval for that row reaches 100 percent, and why up to 15 honest stores reached review in the worst seeds of Table 9.3.

- Labels from reviewed stores alone miscalibrate the model. Table 9.7 shows a milder form of the cold-start failure: the model overstates every store’s probability, and up to 52.3 percent of honest stores get an action in the worst seeds. A fixed warning threshold assumes a calibrated model. Setting it from the distribution of peer scores, such as a high percentile within the store class, would avoid both failures.

External validity. The world has 350 stores, not a state’s thousands. Each simulated basket stands for about 8 real baskets, and shrinkage and feature noise act on the simulated line count. Prices, baskets, store sizes, and most other parameters are assumed, not measured. No real item-level EBT data exist to calibrate against, so the world cannot be checked against reality on the features that matter most. Traffickers follow four fixed strategies, alone or in an even mix. A real adversary searches for the weakest point, and the strategies here are plausible, not optimal. The realistic scenario’s values (detection 0.8, false substantiation 0.01, a quarter of traffickers forging invoices) are assumptions too; this study uses no measured investigation error or forgery rates. The study does not test ALERT or any FNS system.

Construct validity. “Trafficked dollars caught” counts the trafficked dollars held by reviewed stores. In Tables 9.6 to 9.9 it counts them only when the review substantiates the case. Turning that into dollars prevented assumes that a review prevents or recovers the stated 50 percent share. That share and the \$6,000 review cost are the illustrative figures of Section 8.4, not measured ones. Dollars prevented and net benefit inherit both assumptions. Net benefit puts no cost on a wrongful substantiation. Table 9.6 counts those stores but does not price the harm.

10 Trade-offs and harms

Every control in this report imposes costs on honest participants. Table 10.1 lists the main ones (Figure 17).

Table 10.1. Costs and harms of each control, with mitigations

Control	Fraud it targets	Cost or harm	Mitigation
Item-level basket tracking	Overcharging, ineligible sales, fake baskets	Privacy: the state records what poor people eat. Data could be used for purposes beyond integrity, such as food-choice policing	Data minimization, retention limits, statutory purpose limits, aggregation for research
Strict store rules (stocking, scanning)	Front stores, trafficking	Food-desert access loss when small stores leave the program	Subsidized point-of-sale equipment, exceptions for underserved areas, phased deadlines
Payout holds and delays	Hit-and-run trafficking, unrecoverable losses	Honest small stores face cash flow strain	Time limits on holds, partial holds, standard tier for clean history
Aggressive scoring	All store-side fraud	False positives; slow appeals can close an honest store	Appeal deadlines with service levels, independent review, publication of error rates

Control	Fraud it targets	Cost or harm	Mitigation
Heavier eligibility checks	Eligibility fraud, duplicates	Eligible people drop out because of paperwork and delays	Use data matches to reduce paperwork, request documents only on mismatch
Step-up confirmation for large spends	Skimming drains	Delays for recipients without phones	Fail-open path, opt-in, store-based fallback

Privacy. Item-level data is the most sensitive control. Once the state holds a record of every item every SNAP household buys, pressure will follow to use it for diet policing, immigration enforcement, or commercial sale. The data design should limit use by statute to program integrity and approved research. It should keep raw baskets for a short period, such as 13 months, then retain only store-level aggregates. It should separate identity from basket data so that analysts work with tokenized card numbers, and only a case opened by an investigator can resolve a token to a household.

Access. Small stores are often the only food retailers in poor urban and rural areas. They also carry most trafficking. A rule that removes trafficking stores also removes some honest ones, and each loss can strand households far from a supermarket. Mitigation requires subsidy for equipment and an explicit access test before disqualifying the last authorized store in an area, with a path to a civil money penalty instead.

Due process. ALERT cases already rest mostly on transaction data, and stores have ten days to respond to a charge letter [9]. Adding automated holds and caps raises the stakes of a wrong score. The program should publish appeal timelines, track how often automated actions reverse on appeal, and pay interest on wrongful holds.

Human review before cutoff. No household should lose benefits because of an automated score. Scores can open a case. A person decides it, with notice and a hearing, as 7 CFR 273.16 already requires for intentional program violations [8].

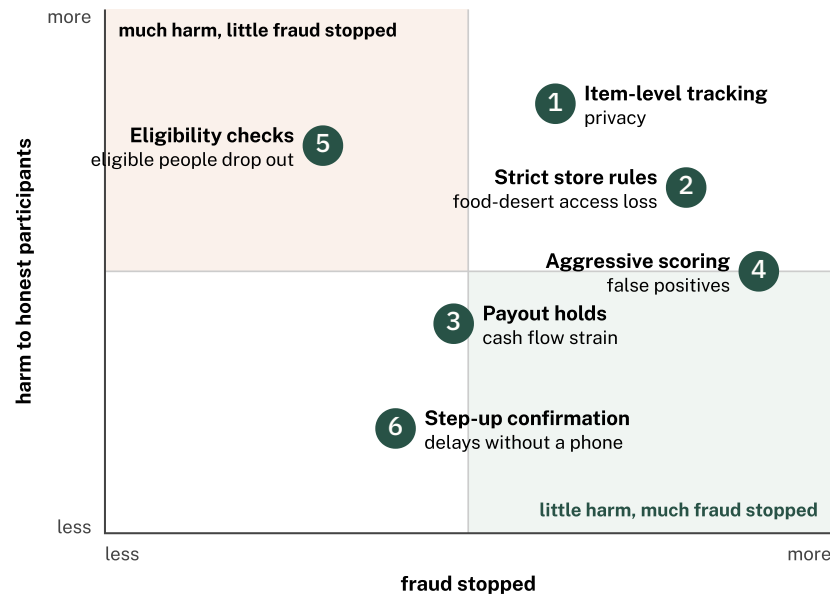
11 The policy alternative

Trafficking exists because SNAP benefits are restricted to food at authorized stores while cash is unrestricted. If the program paid benefits as unrestricted cash, trafficking would end by definition, because there would be no discount between a benefit dollar and a cash dollar for anyone to capture. The retailer authorization system, ALERT, undercover buys, and most of the merchant-side architecture in this report would become unnecessary.

That is a policy choice, and it carries its own debates. Supporters point to lower administrative cost, more dignity and flexibility for households, and economic research finding that households spending more on food than their benefit respond to a dollar of food stamps much as they do to a dollar of cash [26]. Opponents argue that restricting benefits to food is the program’s purpose and the basis of its political support, that cash could shift spending away from food (a retail panel study finds that households spend 50 to 60 cents of each benefit dollar on SNAP-eligible food, far more than out of cash, even when the benefit is below their food spending [24]), and that the change would remove an effect that supports grocery retailers. A middle path, such as a partial cash-out, would shrink the trafficking discount without ending it.

This report takes no position on that question. The point for designers is that the software architecture addresses a problem the policy creates. If policy changes, the integrity problem changes with it: eligibility fraud and payment theft remain, and trafficking disappears.

Qualitative placement: judgment, not measurement



1 Item-level basket tracking

Targets: overcharging, ineligible sales, fake baskets

Harm: privacy: the state records what poor people eat, and could use it for food-choice policing

Mitigation: data minimization, retention limits, statutory purpose limits, aggregation for research

2 Strict store rules

Targets: front stores, trafficking

Harm: food-desert access loss when small stores leave the program

Mitigation: subsidized point-of-sale equipment, exceptions for underserved areas, phased deadlines

3 Payout holds and delays

Targets: hit-and-run trafficking, unrecoverable losses

Harm: honest small stores face cash flow strain

Mitigation: time limits on holds, partial holds, standard tier for clean history

4 Aggressive scoring

Targets: all store-side fraud

Harm: false positives; slow appeals can close an honest store

Mitigation: appeal deadlines with service levels, independent review, published error rates

5 Heavier eligibility checks

Targets: eligibility fraud, duplicates

Harm: eligible people drop out because of paperwork and delays

Mitigation: data matches to cut paperwork, documents only on mismatch

6 Step-up confirmation for large spends

Targets: skimming drains

Harm: delays for recipients without phones

Mitigation: fail-open path, opt-in, store-based fallback

Figure 17: Every control has a cost. Each control stops some fraud and harms some honest participants, and the mitigation decides how much harm remains. **Read it as:** no control sits in the safe corner on its own. The strongest controls carry the most harm, so each one ships with its mitigation or honest households and stores pay for the fraud.

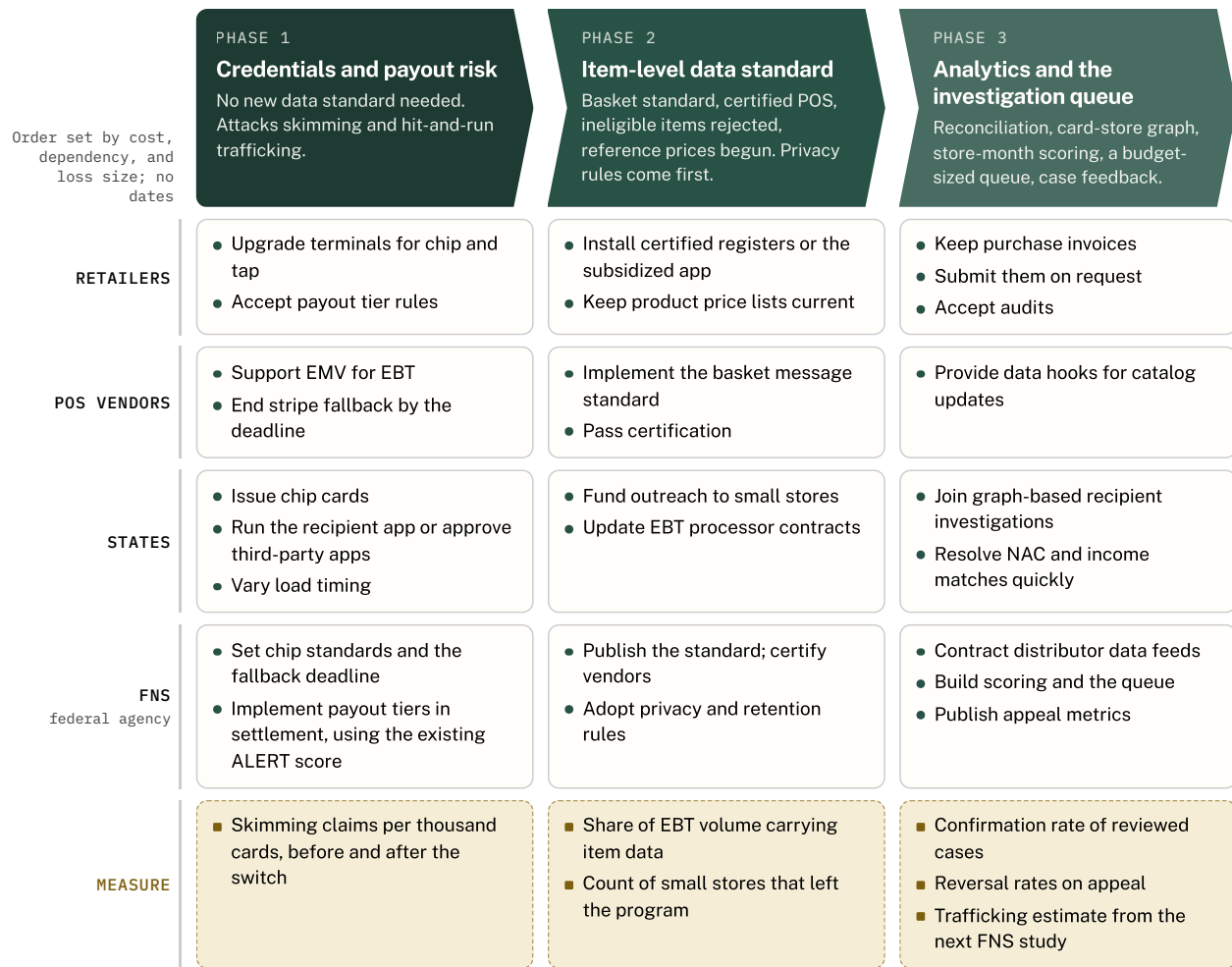


Figure 18: Implementation roadmap: three phases, four parties. Ship the cheap controls that need no new standard first, then item data, then analytics, and measure each phase. **Read it as:** read across a row to see one party's work grow phase by phase, and down a column to see what one phase asks of everyone. Without the measure row, the program cannot tell whether controls pay for themselves.

12 Implementation roadmap

The architecture is too large to deploy at once, and some parts depend on others. A three-phase rollout orders the work by cost, dependency, and the size of the loss each step addresses (Figure 18).

Phase 1: credentials and payout risk. Complete the national move to chip and contactless EBT cards. Launch a recipient app with card lock, geography and online blocks, and per-swipe alerts. Introduce merchant risk tiers using the existing ALERT score: probation for new stores, settlement delays and caps for high tiers. These steps need no new data standard and attack skimming and hit-and-run trafficking directly.

Phase 2: item-level data standard. Define and publish an item-level basket standard for EBT messages. Certify point-of-sale vendors against it. Fund a low-cost certified option for small stores. Turn on real-time rejection of ineligible items. Begin building reference prices and per-store product catalogs. Set privacy rules in regulation before data collection starts.

Phase 3: analytics and the investigation queue. Add inventory reconciliation with dis-

tributor feeds and purchase-record requirements. Build the card-store graph. Deploy store-month price scoring and expected-loss ranking. Size the investigation queue to the budget and run the feedback loop from closed cases.

Table 12.1. Implementation roadmap by phase and actor

Phase	Retailers	Point-of-sale vendors	States	FNS
1	Upgrade terminals for chip and tap; accept payout tier rules	Support EMV for EBT; end stripe fallback by deadline	Issue chip cards; run the recipient app or approve third-party apps; vary load timing	Set chip standards and fallback deadline; implement payout tiers in settlement
2	Install certified registers or the subsidized app; keep product price lists current	Implement the basket message standard; pass certification	Fund outreach to small stores; update EBT processor contracts	Publish the standard; certify vendors; adopt privacy and retention rules
3	Keep purchase invoices; submit them on request; accept audits	Provide data hooks for catalog updates	Join graph-based recipient investigations; resolve NAC and income matches quickly	Contract distributor data feeds; build scoring and the queue; publish appeal metrics

Each phase should ship with measurement. Phase 1 should report skimming claims per thousand cards before and after the switch. Phase 2 should report the share of EBT volume carrying item data and the count of small stores that left the program. Phase 3 should report the confirmation rate of reviewed cases, reversal rates on appeal, and estimated trafficking from the next FNS study. Without measurement, the program cannot tell whether controls pay for themselves.

13 Conclusion

SNAP fraud is smaller and more specific than public debate suggests. The headline error rate near 11 percent measures payment accuracy, and it counts mistakes in both directions. The best estimate of trafficking, the program’s signature fraud, is about 1.6 to 2.0 percent of benefits, concentrated in small stores [50]. Skimming is a separate problem whose victims are recipients.

Trafficking persists because the EBT record carries amounts without items and because both parties to the lie gain from it. Controls that rely on data the shopper or the store enters cannot fully solve it. The architecture in this report responds in layers: verify eligibility from outside sources before the first payment, harden the card, carry items in the transaction, manage payouts the way processors manage risky merchants, and reconcile what stores sell against what their suppliers delivered. Price anomaly detection runs at the store-month level, ranks stores by expected loss, and sends only what the budget covers to people, so that oversight cost tracks investigator headcount. A simulation on synthetic data supports the evidence half of this design more than the ranking half. When traffickers disguise their prices, price features alone catch between 0 and 19.2 percent of trafficked dollars, while reconciliation against supplier purchases keeps capture near 60 percent. That figure needs perfect investigations and honest invoices. Under more realistic assumptions, with imperfect investigations, biased labels, some forged invoices, and a mix of disguises, capture falls to 38.6 percent, and reconciliation still accounts for most of it. Expected-loss ranking and shrinkage showed no measurable gain over simpler choices in that setting.

Each control harms some honest participants, through lost privacy, lost store access, delayed payouts, or wrongful flags. Those costs belong in the same calculation as the fraud the controls stop. A program that counts them, publishes its error and reversal rates, and keeps a person between any score and a family's food will protect both the budget and the trust that the program depends on.

Ethics statement

This study uses synthetic data only. No personal data, household data, retailer data, or EBT transaction records were collected or used. The simulated stores and households are generated by code published with the paper, and no simulated store stands for a real one.

The work carries a dual-use risk. A description of how stores are scored also tells a trafficking store how to evade the score, for example by keeping every price inside the reference range. The paper limits that risk in two ways. It publishes the method and not any real thresholds or model weights. And its architecture rests on evidence a store does not write, such as distributor records of what suppliers delivered, so that knowing the score does not make evasion free. Invoices a store submits itself can be forged, and Table 9.8 shows what that costs.

False flags carry real harm. A wrongly flagged small store can face payout delays, volume caps, or disqualification. The households that shop there can lose the nearest place to use their benefits. Those costs fall on people with little cushion, and the simulation reports them as a result, not a footnote. The design keeps a person between any score and a sanction: scores open cases, people decide them, and every automated action expires and can be appealed (Sections 6 and 10).

A UML models

These models turn the design into operations and software specifications. The operations models come first.

A.1 Operations

The figures so far follow the money and the fraud. The next four show the operations side in UML, the notation software teams use to specify systems. The first is a use case diagram. It names each party that touches the integrity program and what each one does there, from the household and the store to the federal retailer analyst at FNS (renamed FNA in 2026).

Two points stand out. Recipients and retailers run controls themselves: a household locks its own card, and a store submits its own purchase records. And every sanction rests with a person, as Section 10 requires.

An activity diagram shows how one case moves between parties. It starts in software. The analytics system scores every store each month, and only stores in the top N reach a federal analyst; the rest get an automated warning letter, payout delay, or volume cap. Once a case opens, three lines of evidence run at the same time: supplier records, an undercover buy, and the store's own purchase records.

The case file joins them before anyone decides. A sanction can go to independent review, and every outcome, closed, upheld, or reversed, becomes a label for next month's scores.

The same process, seen from the case record, is a state machine. Each box is a status the case can hold, and each arrow names the event, guard, and action that move it. The design rules from Sections 7, 8, and 10 become guards here.

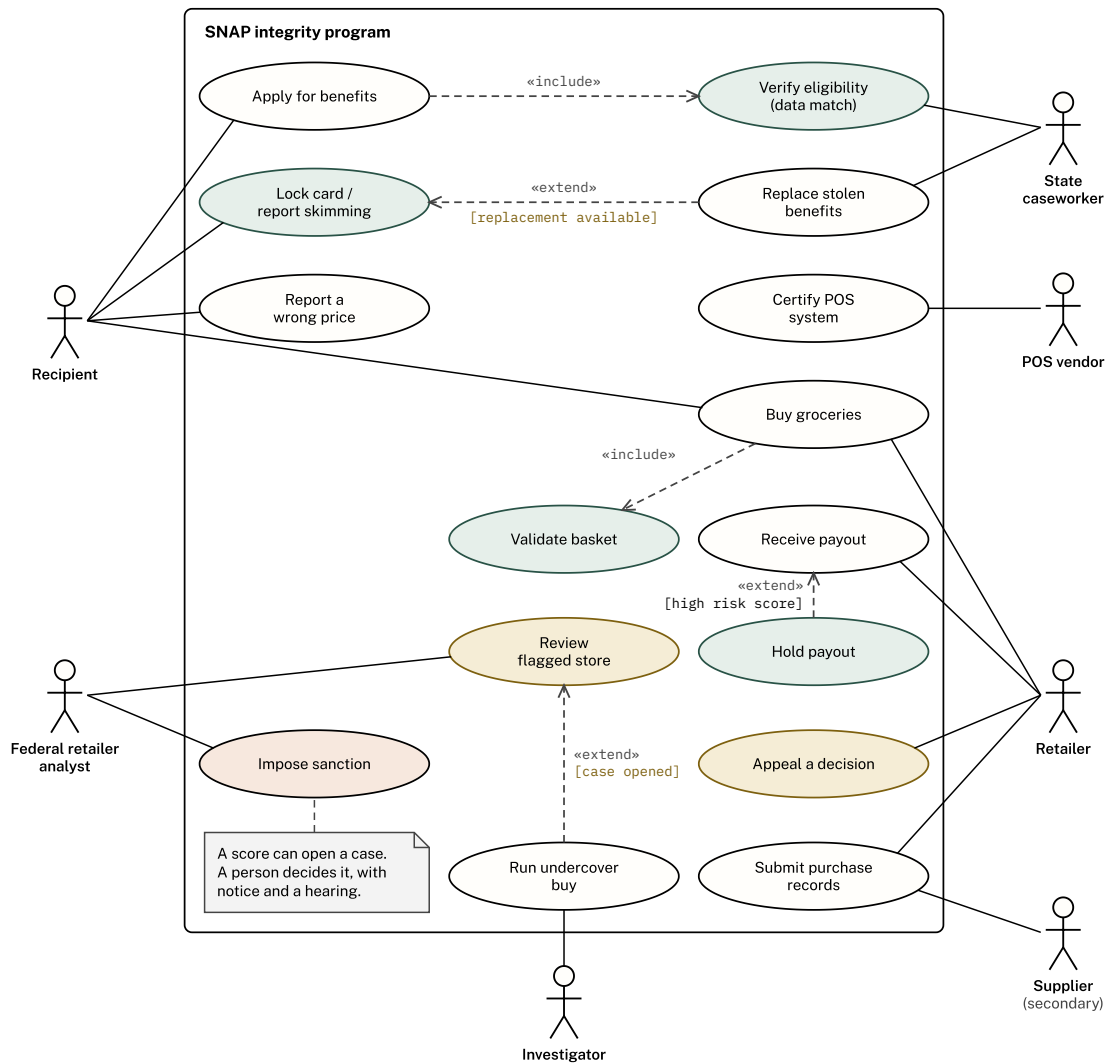


Figure A1: Who does what in the integrity program. Recipients and retailers use the program; caseworkers, analysts, and investigators police it. A person decides every sanction. **Read it as:** stick figures are actors outside the system, and ovals are use cases inside it. A solid line means the actor takes part. A dashed «include» arrow points to a step its base case always runs. A dashed «extend» arrow points back to the base case it sometimes adds to, under the condition in brackets.

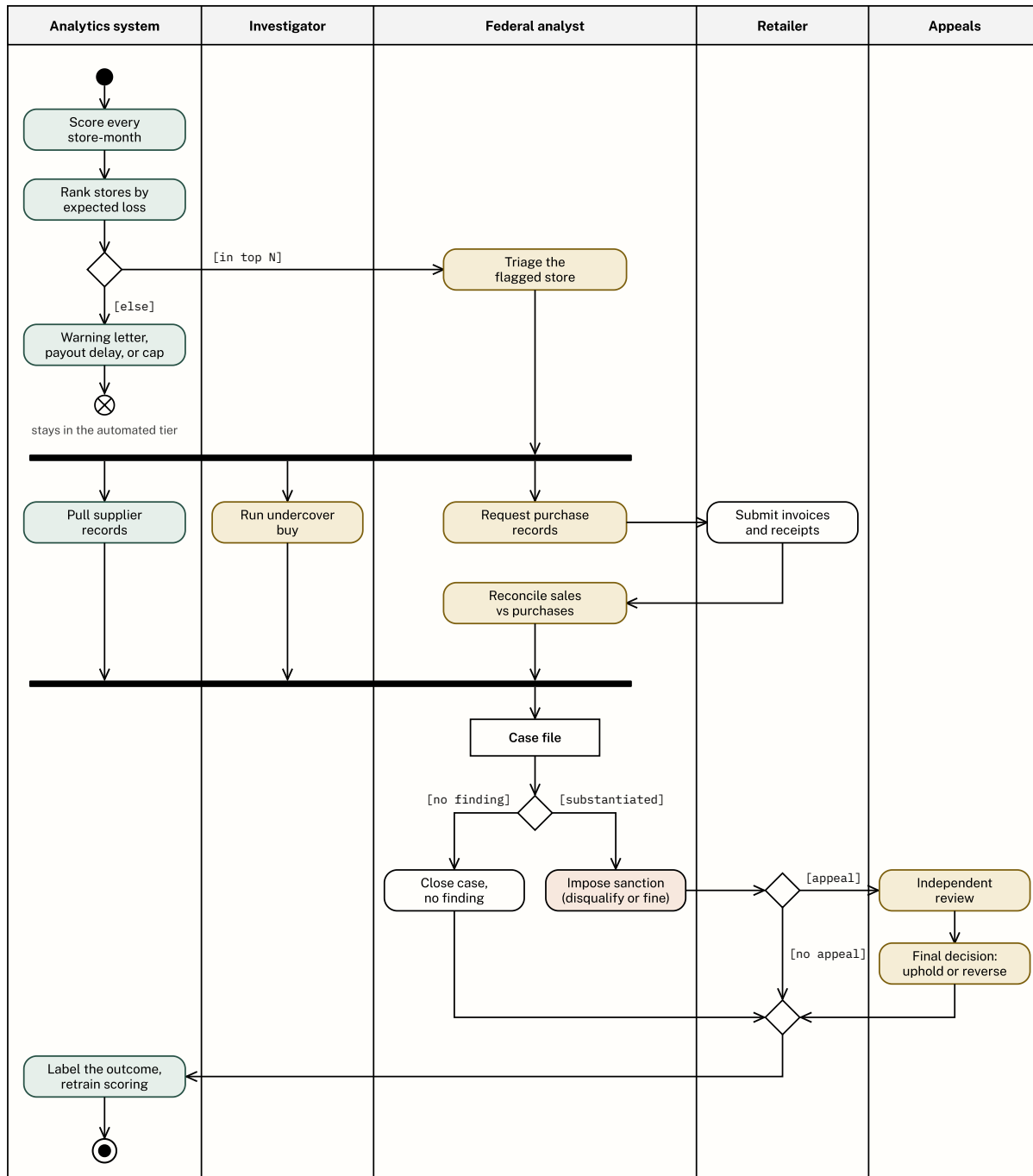


Figure A2: From flagged store to final decision. Software picks the store, three lines of evidence run in parallel, and every closed case returns to scoring as a label. **Read it as:** each column is one party. The black dot starts the flow, and the bullseye ends it. Bars split work into parallel branches and join it again. Diamonds pick one branch by the guard in brackets, or merge branches back into one. The circled X marks where a store below the top N leaves the flow without reaching a person.

A case opens only when its expected loss beats the review cost within budget. A payout hold starts when the case enters the queue and lifts if the evidence shows no violation. Every sanction can go to appeal before the case becomes final.

The last diagram turns to the recipient side, where the loss is theft from families. Section 7.2 proposes a recipient app with per-swipe alerts and a card lock, and Section 5.4 describes how benefit replacement worked. The activity diagram joins the two.

A family can stop a drain within minutes of the first unknown swipe. The response then splits: the state reissues the card and decides whether the stolen benefits come back. Under the facts in Section 5.4, federal replacement covered thefts only through December 20, 2024.

A.2 Software

The component diagram recasts the layered design as software. Each box is a deployable component. A ball marks an interface the component offers, and a cup marks one it needs. The real-time path answers inside a sale: the gateway calls the eligibility, credential, and basket checks, then settles through the payout risk service.

Batch analytics reads the stream of sales once a month. Three evidence services feed one scoring engine. External sources sit outside the packages, because the program reads them and cannot change them.

The sequence diagram follows one EBT purchase once the basket travels with the authorization. The processor checks every line against SNAP eligibility and the store catalog before it touches the balance. A valid basket then passes a risk check. Only a large spend soon after the monthly load triggers a step-up confirmation in the recipient app, and a recipient without a phone fails open.

An invalid basket is declined at the register. The processor still reports the line, because a store that keeps keying items it never stocks is a signal for the monthly score.

The class diagram names the records the software keeps. Two structures mirror each other: a transaction owns its basket lines, and a supplier invoice owns its invoice lines. Both kinds of line point at the same product barcode. That shared key makes reconciliation possible, because the store writes one side and its suppliers write the other.

Scores attach to a retailer and a month, never to a single basket. A score can open a case or trigger an automatic action, and every action carries an expiry and an appeal.

The second sequence diagram shows the batch side. A scheduler starts the run once a month. For each store, the scoring engine gathers three kinds of evidence: excess prices against the regional reference, sales against supplier purchases, and the store's place in the card-store graph. It scores the store-month with shrinkage, which is meant to keep a small store with a few odd lines from jumping the queue. The simulation in Section 9 finds no measurable effect from this step in its setting.

The ranker sorts stores by expected loss. The investigator budget fixes N before the run starts, so transaction volume never changes how many cases people see.

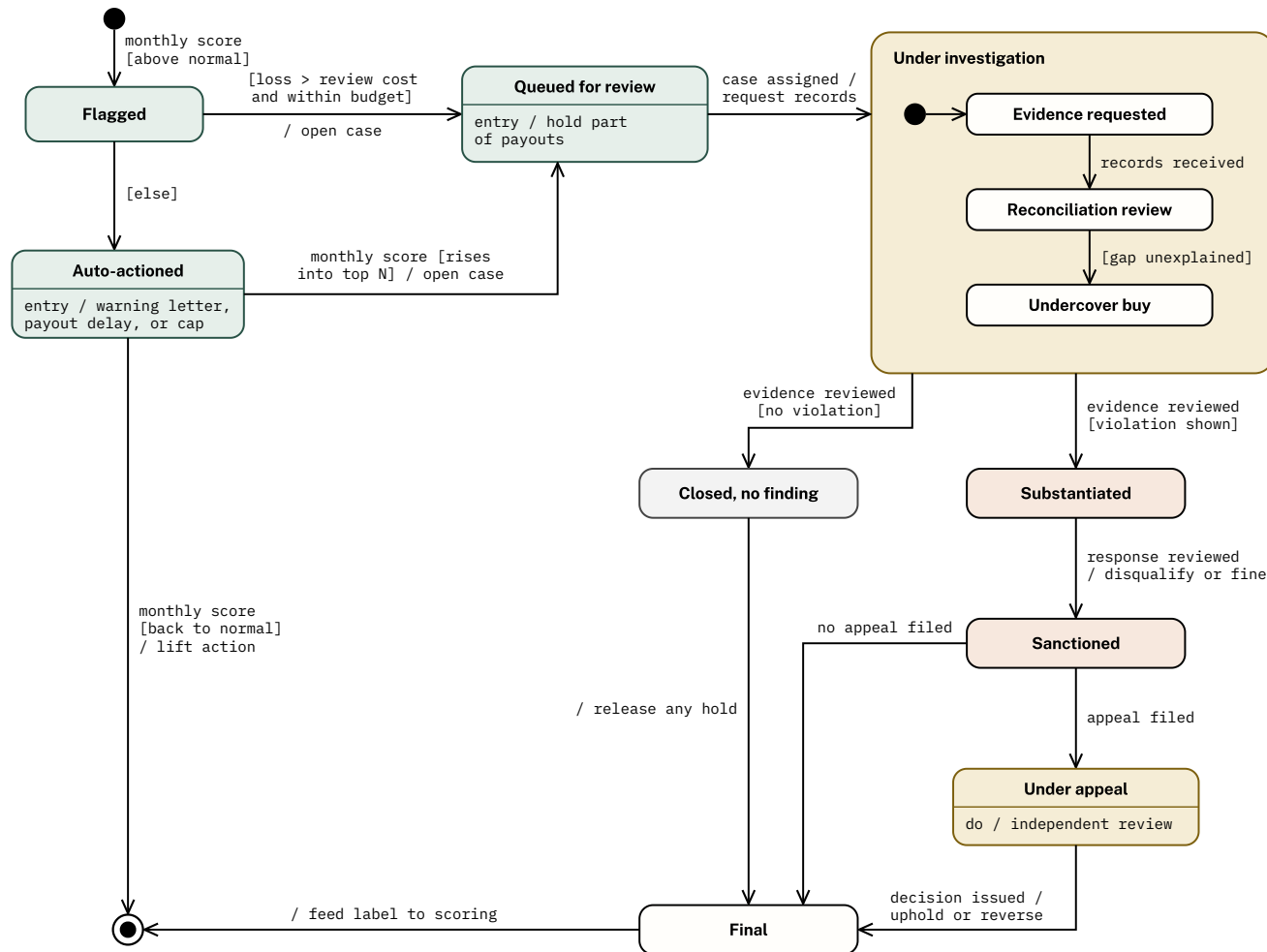


Figure A3: The life of an integrity case. A score can open a case and hold some payouts. Only evidence moves it to a sanction, and every sanction can go to appeal. **Read it as:** rounded boxes are states, and arrows are transitions labeled event [guard] / action. The large box groups three investigation substates, and its two exits fire from any of them. An entry line runs when the case enters that state; a do line runs while it stays there.

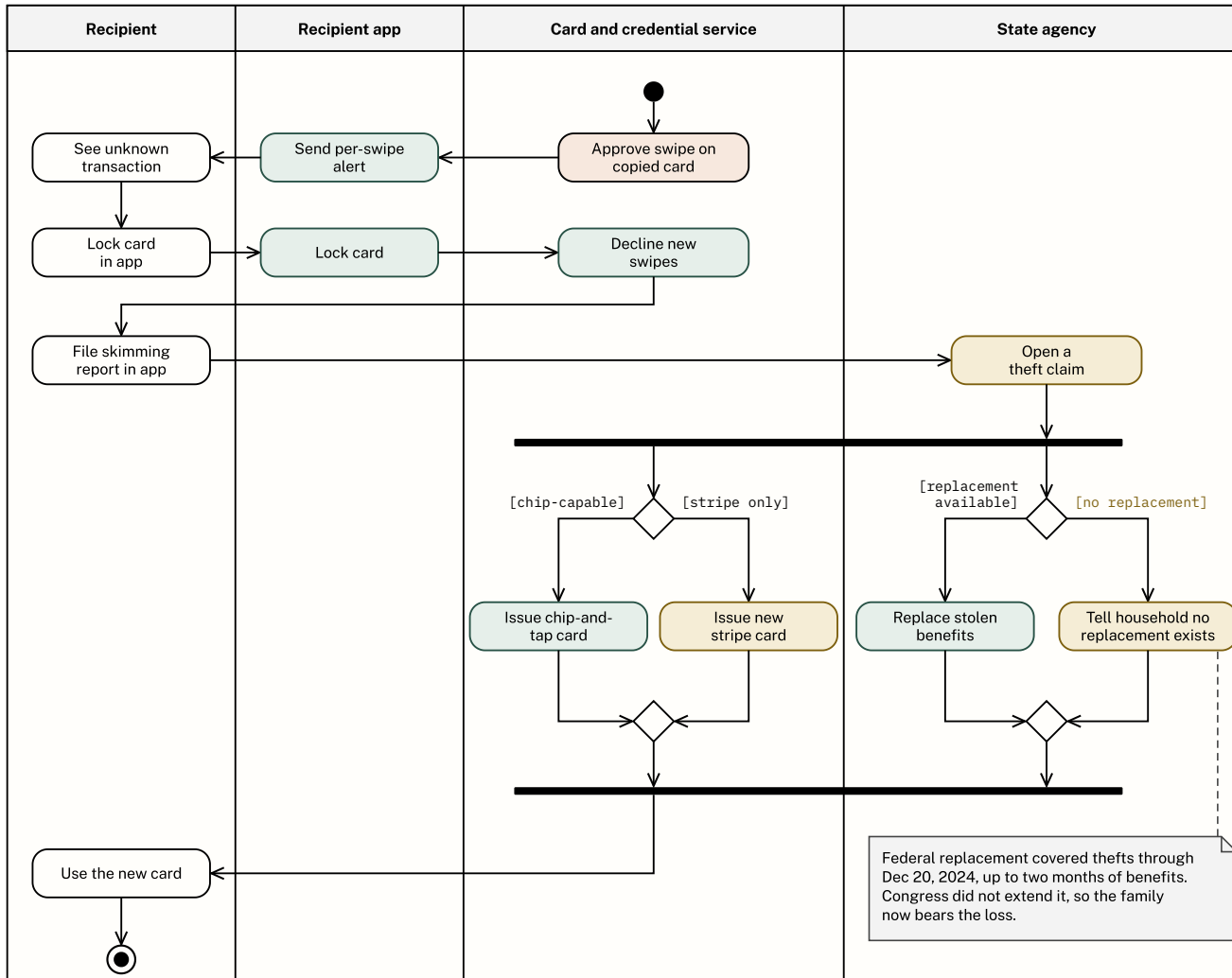


Figure A4: When a skimmed card gets drained. The app lets a family stop a drain within minutes. Whether the stolen money comes back depends on when the theft happened. **Read it as:** each column is one party. After the claim opens, two branches run in parallel: a new card, and the question of the stolen benefits. A chip card blocks the next copy; a new stripe card can be copied again. The note gives the report's facts on federal replacement.

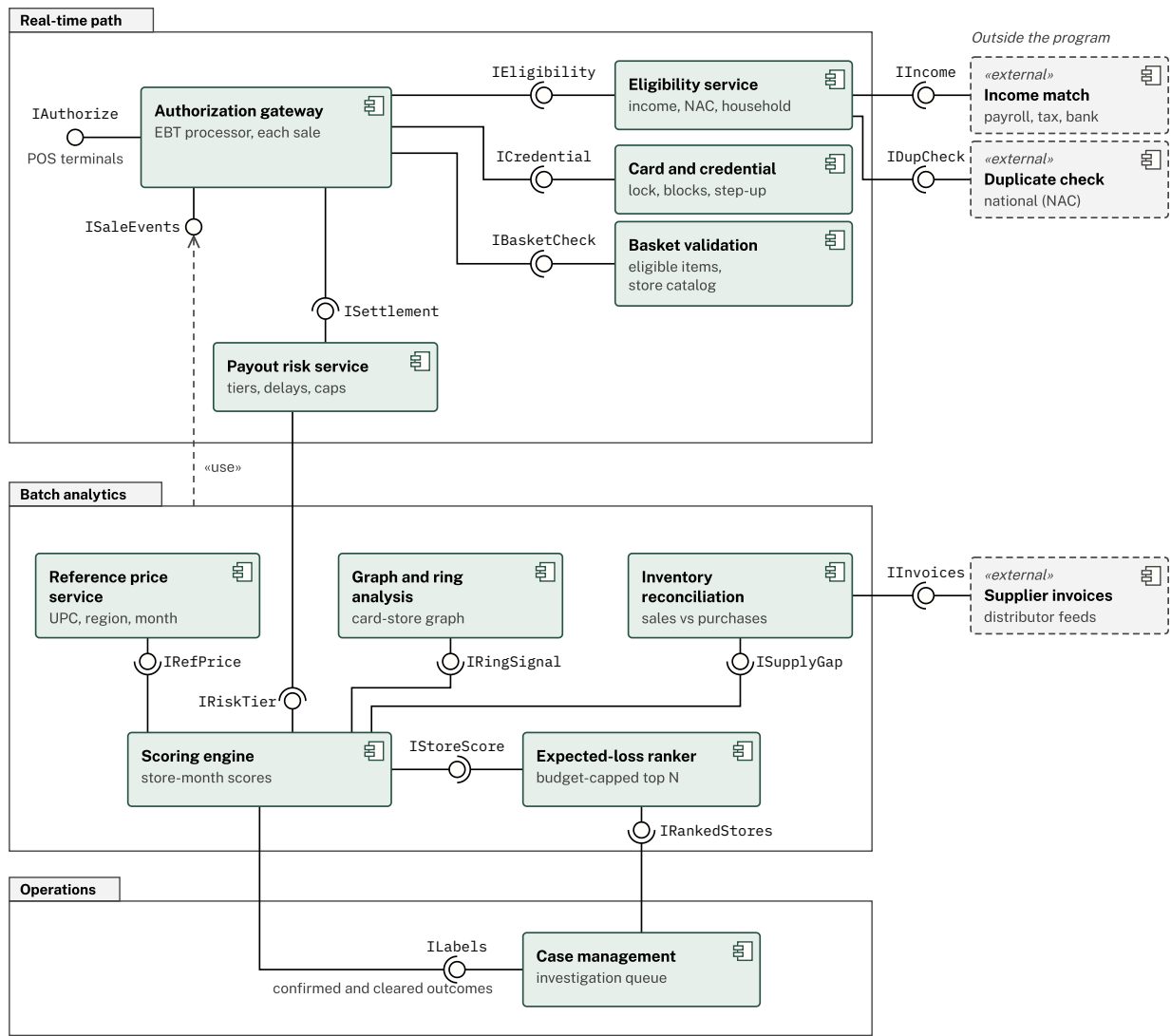


Figure A5: The platform as components. Real-time services guard each sale, batch analytics scores stores each month, and case outcomes flow back as labels. **Read it as:** a ball is an interface a component provides, and a cup is one it requires. The gateway calls three checks on every sale and settles through payout risk. Only two outputs leave the scoring engine: a risk tier for payouts and scores for the ranker. People see only the ranked list, and their outcomes retrain the engine.

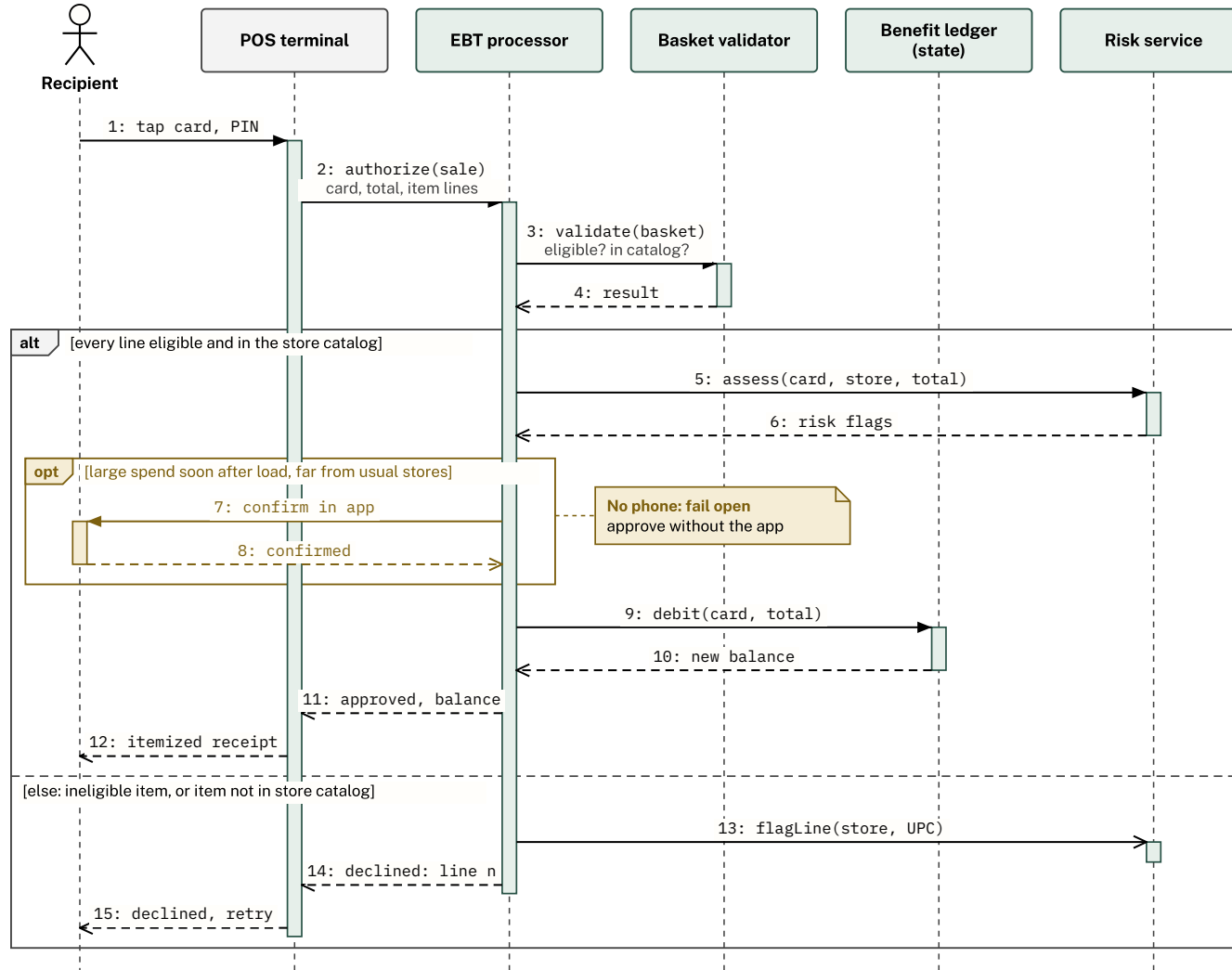


Figure A6: One purchase, line by line. The processor checks every basket line before it touches the balance, and asks the recipient only when a spend looks like a load-day drain. **Read it as:** solid arrows with filled heads are calls, dashed arrows are replies, and the open-headed arrow is a one-way signal. The alt frame runs exactly one branch; the opt frame runs only when its guard holds. A rejected line ends the sale before any money moves, and the risk service still hears about it.

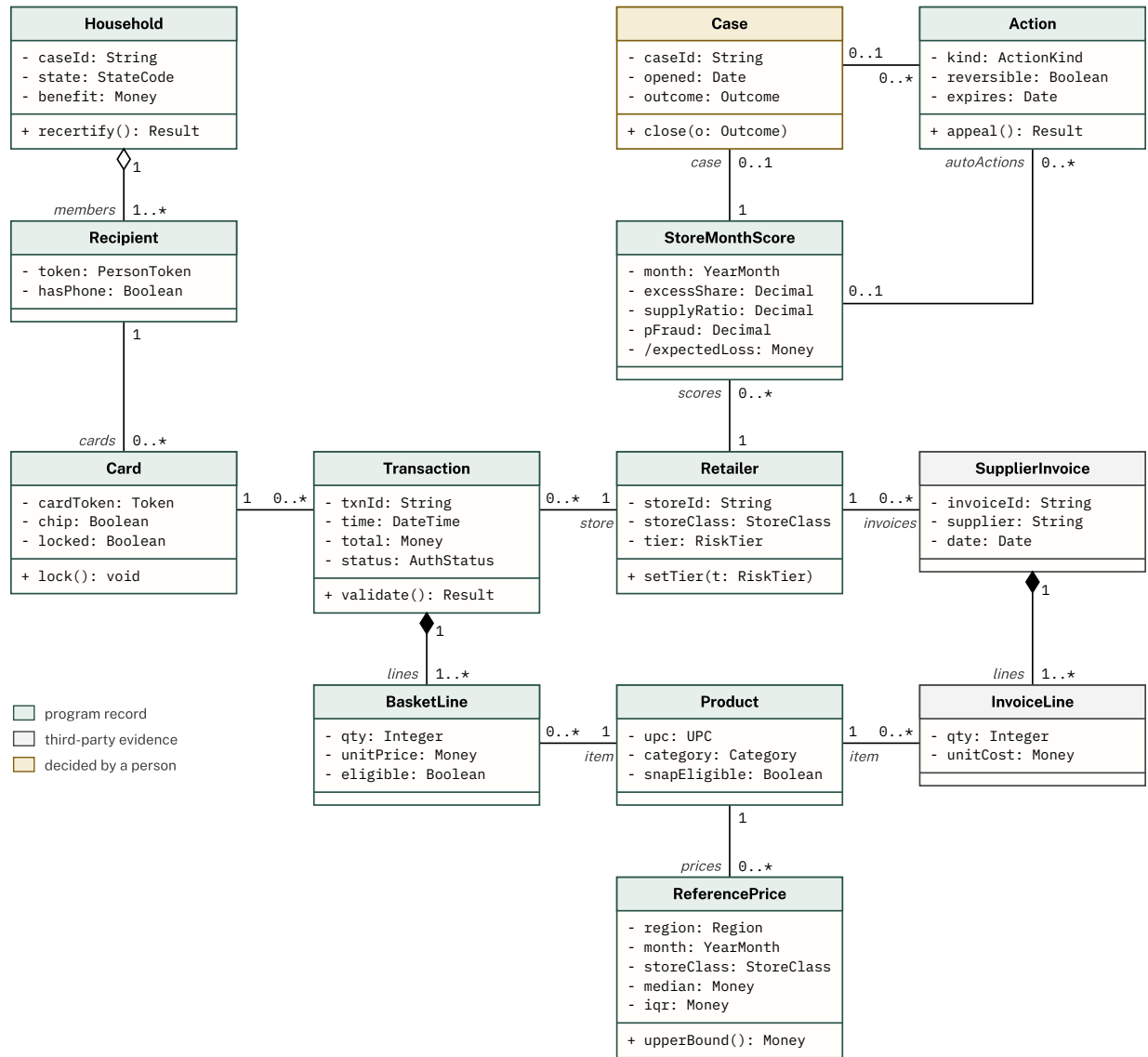


Figure A7: The domain model. Thirteen classes carry the design: sales with their lines, purchases with theirs, and scores that open cases. **Read it as:** a filled diamond marks a part that lives and dies with its whole, so a basket line exists only inside its transaction. Numbers at each end are multiplicities. Product is the hub: basket lines, invoice lines, and reference prices all point at one barcode, and that shared key lets sales be reconciled against purchases.

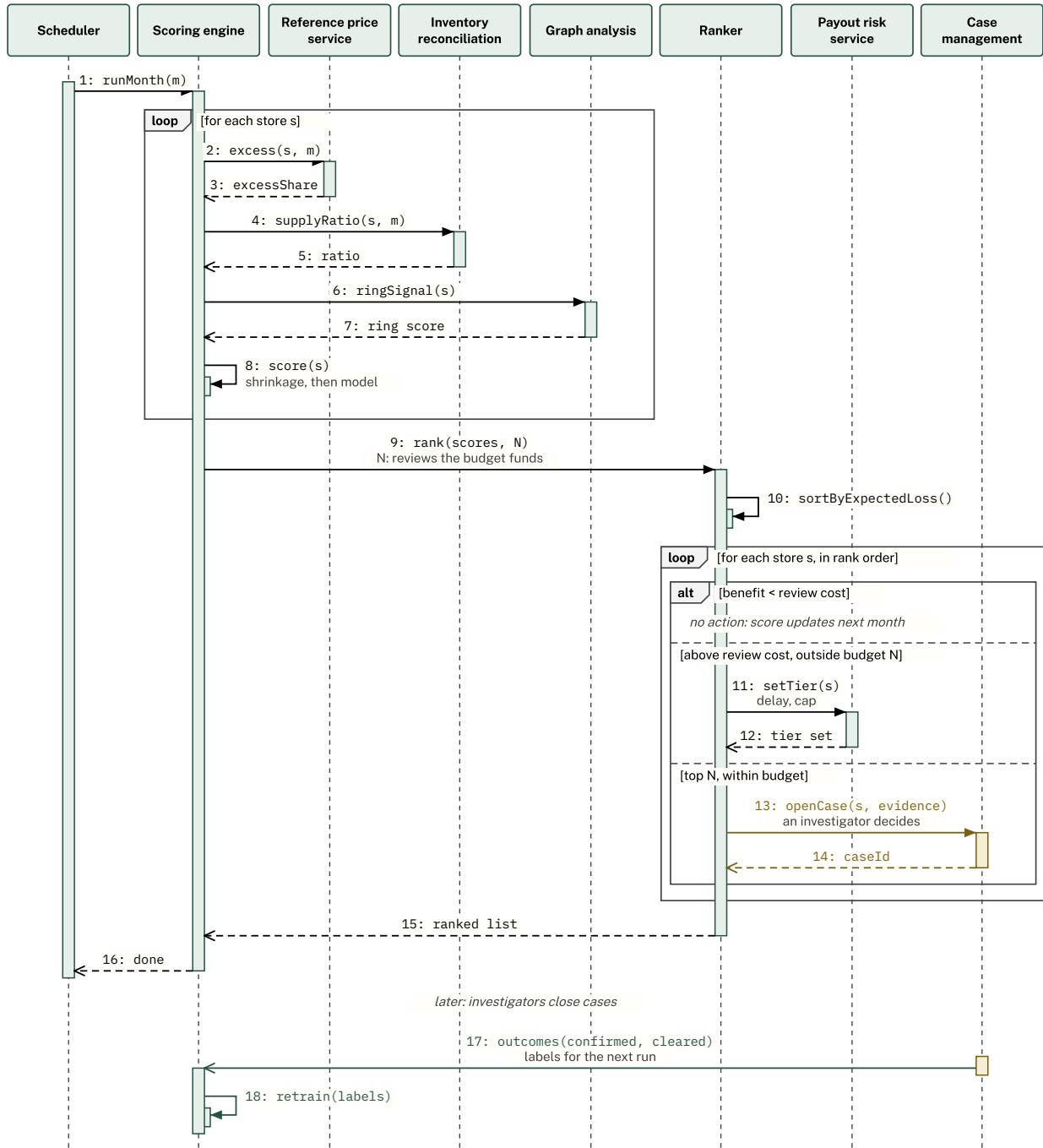


Figure A8: The monthly scoring run. Each month the engine scores every store, the ranker sorts them by expected loss, and the budget decides who gets a person. **Read it as:** the loop frames repeat once per store, and the alt frame picks one branch per store. Only the top N within budget reach case management. The rest get a reversible automatic action or nothing. Closed cases come back weeks later as labels, and the next run trains on them.

- [1] Agriculture Improvement Act of 2018. Public Law 115-334, section 4006(e), 132 Stat. 4490, 4635-4636, December 20, 2018. [congress.gov: P.L. 115-334](https://www.congress.gov/115/legislation/115-334)
- [2] Biggio, Battista, and Fabio Roli. “Wild Patterns: Ten Years After the Rise of Adversarial Machine Learning.” *Pattern Recognition* 84 (2018): 317-331. [doi.org: 10.1016/j.patcog.2018.07.023](https://doi.org/10.1016/j.patcog.2018.07.023)
- [3] Black, Emily, Hadi Elzayn, Alexandra Chouldechova, Jacob Goldin, and Daniel E. Ho. “Algorithmic Fairness and Vertical Equity: Income Fairness with IRS Tax Audit Models.” In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, 1479-1503. 2022. [doi.org: 10.1145/3531146.3533204](https://doi.org/10.1145/3531146.3533204)
- [4] Bolton, Richard J., and David J. Hand. “Statistical Fraud Detection: A Review.” *Statistical Science* 17, no. 3 (2002): 235-255. [doi.org: 10.1214/ss/1042727940](https://doi.org/10.1214/ss/1042727940)
- [5] *Cahoo v. SAS Analytics Inc.*, 912 F.3d 887 (6th Cir. 2019). US Court of Appeals for the Sixth Circuit, January 3, 2019. [courtlistener.com: Cahoo v. SAS Analytics](https://www.courtlistener.com/cases/circuit/circuit-6/circuit-6-19-0001/)
- [6] Center on Budget and Policy Priorities. “Congress Must Extend Protections for SNAP Households Who Are Victims of Benefit Theft.” Blog post, 2024. [cbpp.org: SNAP benefit theft](https://www.cbpp.org/snap-benefit-theft)
- [7] Chandola, Varun, Arindam Banerjee, and Vipin Kumar. “Anomaly Detection: A Survey.” *ACM Computing Surveys* 41, no. 3 (2009): article 15, 1-58. [doi.org: 10.1145/1541880.1541882](https://doi.org/10.1145/1541880.1541882)
- [8] Code of Federal Regulations. 7 CFR 273.16, Disqualification for intentional Program violation. Accessed via Legal Information Institute, Cornell Law School. [law.cornell.edu: 7 CFR 273.16](https://www.legis.law.cornell.edu/cfr/text/7/273.16)
- [9] Code of Federal Regulations. 7 CFR 278.6, Disqualification of retail food stores and wholesale food concerns and imposition of civil money penalties in lieu of disqualifications. Accessed via Legal Information Institute, Cornell Law School. [law.cornell.edu: 7 CFR 278.6](https://www.legis.law.cornell.edu/cfr/text/7/278.6)
- [10] Congressional Research Service. *Supplemental Nutrition Assistance Program: Errors and Fraud* (IF10860). Updated April 7, 2025. [everycrsreport.com: IF10860](https://www.everycrsreport.com/reports/IF10860/)
- [11] Congressional Research Service. *Errors and Fraud in the Supplemental Nutrition Assistance Program (SNAP)* (R45147). Updated September 28, 2018. [everycrsreport.com: R45147](https://www.everycrsreport.com/reports/R45147/)
- [12] Congressional Research Service. *Supplemental Nutrition Assistance Program (SNAP): Benefit Theft Through Electronic Benefit Card Skimming* (IN12419). [congress.gov: IN12419](https://www.congress.gov/116/legislation/116-12419)
- [13] Consolidated Appropriations Act, 2023. Public Law 117-328, Division HH, section 501.
- [14] Contra Costa County Employment and Human Services Department. “New EBT Card Security & Technology Upgrade.” February 2025. [ehsd.org: EBT card upgrade](https://ehsd.org/ebt-card-upgrade)
- [15] Dal Pozzolo, Andrea, Giacomo Boracchi, Olivier Caelen, Cesare Alippi, and Gianluca Bontempi. “Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy.” *IEEE Transactions on Neural Networks and Learning Systems* 29, no. 8 (2018): 3784-3797. [doi.org: 10.1109/TNNLS.2017.2736643](https://doi.org/10.1109/TNNLS.2017.2736643)
- [16] Dalvi, Nilesh, Pedro Domingos, Mausam, Sumit Sanghai, and Deepak Verma. “Adversarial Classification.” In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 99-108. 2004. [doi.org: 10.1145/1014052.1014066](https://doi.org/10.1145/1014052.1014066)
- [17] Efron, Bradley, and Carl Morris. “Data Analysis Using Stein’s Estimator and Its Generalizations.” *Journal of the American Statistical Association* 70, no. 350 (1975): 311-319. [doi.org: 10.1080/01621459.1975.10479864](https://doi.org/10.1080/01621459.1975.10479864)
- [18] Elkan, Charles. “The Foundations of Cost-Sensitive Learning.” In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI)*, 973-978. 2001. [dl.acm.org: 10.5555/1642194.1642224](https://dl.acm.org/doi/10.5555/1642194.1642224)
- [19] Eubanks, Virginia. *Automating Inequality: How High-Tech Tools Profile, Police, and Punish the Poor*. New York: St. Martin’s Press, 2018. ISBN 978-1-250-07431-7. [openlibrary.org: Automating Inequality](https://openlibrary.org/subjects/automating-inequality)
- [20] Federal Register. “Enhancing Retailer Standards in the Supplemental Nutrition Assistance Program (SNAP).” Final rule, 81 FR 90675, December 15, 2016. [federalregister.gov: 81 FR 90675](https://www.federalregister.gov/2016/12/15/90675)
- [21] Federal Register. “Supplemental Nutrition Assistance Program: Requirement for Interstate Data Matching To Prevent Duplicate Issuances.” Interim final rule, October 3, 2022. [federalregister.gov: interstate data matching](https://www.federalregister.gov/2022/10/03/interstate-data-matching)
- [22] Federal Register. “Updated Staple Food Stocking Standards for Retailers in the Supplemental Nutrition Assistance Program.” Final rule, FR Doc. 2026-09137, May 8, 2026. [federalregister.gov: 2026-09137](https://www.federalregister.gov/2026/05/08/2026-09137)

- [23] Glaeser, Edward L., Andrew Hillis, Scott Duke Kominers, and Michael Luca. “Crowdsourcing City Government: Using Tournaments to Improve Inspection Accuracy.” *American Economic Review* 106, no. 5 (2016): 114-118. [doi.org: 10.1257/aer.p20161027](https://doi.org/10.1257/aer.p20161027)
- [24] Hastings, Justine, and Jesse M. Shapiro. “How Are SNAP Benefits Spent? Evidence from a Retail Panel.” *American Economic Review* 108, no. 12 (2018): 3493-3540. [doi.org: 10.1257/aer.20170866](https://doi.org/10.1257/aer.20170866)
- [25] Hodges, Leslie, and Jordan W. Jones. *The Food and Nutrition Assistance Landscape: Fiscal Year 2025 Annual Report* (EIB-304). US Department of Agriculture, Economic Research Service, June 2026. [ers.usda.gov: EIB-304](https://ers.usda.gov/EIB-304)
- [26] Hoynes, Hilary W., and Diane Whitmore Schanzenbach. “Consumption Responses to In-Kind Transfers: Evidence from the Introduction of the Food Stamp Program.” *American Economic Journal: Applied Economics* 1, no. 4 (2009): 109-139. [doi.org: 10.1257/app.1.4.109](https://doi.org/10.1257/app.1.4.109)
- [27] James, W., and Charles Stein. “Estimation with Quadratic Loss.” In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, 361-379. Berkeley: University of California Press, 1961. [projecteuclid.org: bsmmsp/1200512173](https://projecteuclid.org:bsmsp/1200512173)
- [28] Kumar, Mohit, Rayid Ghani, and Zhu-Song Mei. “Data Mining to Predict and Prevent Errors in Health Insurance Claims Processing.” In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 65-74. 2010. [doi.org: 10.1145/1835804.1835816](https://doi.org/10.1145/1835804.1835816)
- [29] Li, Jing, Kuei-Ying Huang, Jionghua Jin, and Jianjun Shi. “A Survey on Statistical Methods for Health Care Fraud Detection.” *Health Care Management Science* 11, no. 3 (2008): 275-287. [doi.org: 10.1007/s10729-007-9045-4](https://doi.org/10.1007/s10729-007-9045-4)
- [30] Massachusetts Law Reform Institute. “Congress’s Last-Minute Compromise to Avoid a Shutdown Eliminated its Previous Agreement to Replace Food Benefits Stolen from Low-Income Families.” Blog post, December 23, 2024. [mlri.org: stolen benefits replacement](https://mlri.org:stolen-benefits-replacement)
- [31] One Big Beautiful Bill Act. Public Law 119-21, section 10105, July 4, 2025. [congress.gov: P.L. 119-21](https://congress.gov:P.L.119-21)
- [32] Rekenkamer Rotterdam (Rotterdam Court of Audit). *Gekleurde technologie: verkenning ethisch gebruik algoritmes*. April 15, 2021. [rekenkamers.nl: Gekleurde technologie](https://rekenkamers.nl:Gekleurde-technologie)
- [33] Royal Commission into the Robodebt Scheme (Commissioner Catherine Holmes). *Report*. Commonwealth of Australia, July 7, 2023. [robodebt.royalcommission.gov.au: report](https://robodebt.royalcommission.gov.au:report)
- [34] US Department of Agriculture, Economic Research Service. “Supplemental Nutrition Assistance Program (SNAP) average monthly participation and inflation-adjusted annual program spending, fiscal years 2000-24.” Chart. [ers.usda.gov: participation and spending chart](https://ers.usda.gov:participation-and-spending-chart)
- [35] US Department of Agriculture, Food and Nutrition Service. *Break the Rules, Pay the Price: FNS Retailer Oversight, Fiscal Year 2025*. Infographic, 2026. [fns.usda.gov: retailer oversight FY 2025](https://fns.usda.gov:retailer-oversight-FY-2025)
- [36] US Department of Agriculture, Food and Nutrition Service. *Feasibility Study of Capturing Supplemental Nutrition Assistance Program (SNAP) Purchases at the Point of Sale*. Prepared by IMPAQ International. November 2016.
- [37] US Department of Agriculture, Food and Nutrition Service. *Fiscal Year 2024 SNAP Quality Control Payment Error Rates*. 2025. [fns.usda.gov: FY 2024 QC report](https://fns.usda.gov:FY-2024-QC-report)
- [38] US Department of Agriculture, Food and Nutrition Service. *SNAP Retailer Management Year End Summary, Fiscal Year 2021*. [fns.usda.gov: FY 2021 retailer summary](https://fns.usda.gov:FY-2021-retailer-summary)
- [39] US Department of Agriculture, Food and Nutrition Service. “USDA Releases Annual SNAP Payment Error Rates for FY 2024.” Press release, June 27, 2025.
- [40] US Department of Agriculture, Office of Inspector General. *Detecting Potential SNAP Trafficking Using Data Analysis* (Audit Report 27901-0002-13). January 2017. [oig.usda.gov: 27901-0002-13](https://oig.usda.gov:27901-0002-13)
- [41] US Department of Agriculture. “USDA Announces FY 2025 State Payment Error Rates in SNAP.” Press release, June 24, 2026. [usda.gov: FY 2025 state error rates](https://usda.gov:FY-2025-state-error-rates)
- [42] US Department of Agriculture. “USDA Announces Actions to Better Serve States, Nutrition Program Recipients, and the American Taxpayer.” Press release on the Food and Nutrition Administration, April 30, 2026. [usda.gov: FNS restructuring](https://usda.gov:FNS-restructuring)
- [43] US Department of Justice, US Attorney’s Office, District of Kansas. “Store Owner Sentenced To Prison In Cash-For-Food-Stamps Case.” Press release, March 13, 2012. [justice.gov: District of Kansas](https://justice.gov:District-of-Kansas)
- [44] US Department of Justice, US Attorney’s Office, District of New Jersey. “Camden Man Sentenced To 37 Months In Prison For Exchanging More Than \$2.5 Million In Snap/Food Stamp Benefits For Cash.” Press release, June 5, 2014. [justice.gov: District of New Jersey](https://justice.gov:District-of-New-Jersey)

- [45] US Department of Justice, US Attorney’s Office, Southern District of West Virginia. “Former Huntington Convenience Store Manager Sentenced To Federal Prison For Food Stamp Fraud.” Press release, October 3, 2012. [justice.gov: Southern District of West Virginia](https://www.justice.gov/sdovw)
- [46] US Government Accountability Office. *Food Stamp Trafficking: FNS Could Enhance Program Integrity by Better Targeting Stores Likely to Traffic and Increasing Penalties* (GAO-07-53). October 2006. [gao.gov: GAO-07-53](https://www.gao.gov/products/GAO-07-53)
- [47] US Government Accountability Office. *Medicare: CMS Fraud Prevention System Uses Claims Analysis to Address Fraud* (GAO-17-710). August 2017. [gao.gov: GAO-17-710](https://www.gao.gov/products/GAO-17-710)
- [48] US Government Accountability Office. *Supplemental Nutrition Assistance Program: Actions Needed to Better Measure and Address Retailer Trafficking* (GAO-19-167). December 2018. [gao.gov: GAO-19-167](https://www.gao.gov/products/GAO-19-167)
- [49] van Bekkum, Marvin, and Frederik Zuiderveen Borgesius. “Digital Welfare Fraud Detection and the Dutch SyRI Judgment.” *European Journal of Social Security* 23, no. 4 (2021): 323-340. [doi.org: 10.1177/13882627211031257](https://doi.org/10.1177/13882627211031257)
- [50] Wilson, Hoke (Manhattan Strategy Group). *The Extent of Trafficking in the Supplemental Nutrition Assistance Program: 2015-2017*. USDA Food and Nutrition Service, Office of Policy Support, September 2021. [fns.usda.gov: trafficking 2015-2017](https://fns.usda.gov/trafficking-2015-2017)