

bbox: a mailbox with many keepers and a payment inside

A mailbox someone else runs can lose your mail, show it to others, or claim it never arrived, and you cannot tell which. bbox gives a box several independent keepers and lets you check each of them. Envelopes are sealed to one recipient, bound to their sender, priced by postage built into their structure, delivered once to every keeper by BSV Layered Multicast, and retired everywhere by one signed receipt. A payment can travel inside the envelope and settle when the recipient claims it.

Author: Jeff Harris · Lightweb Inc. · jeff@lightweb.net

Pattern 002 · Version 01

Audience: application developers, founders and business leaders. No Bitcoin or networking background assumed; every term is explained on first use.

We = Lightweb Inc., designer of the bstack patterns and operator of the 1BSV network; **you** = a builder or decision maker evaluating them.

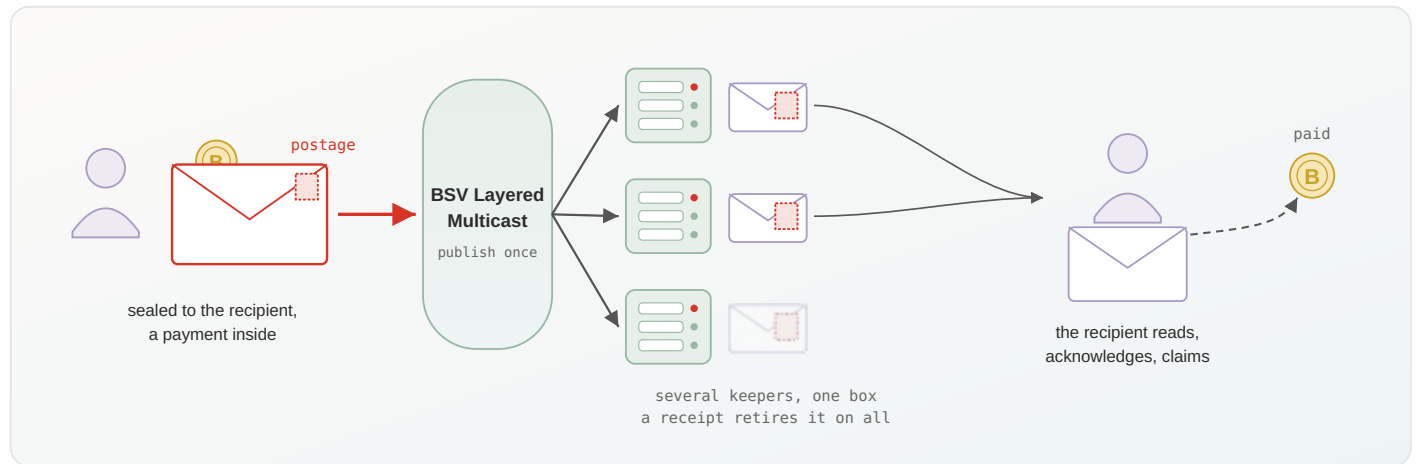


Figure 1: one envelope, several keepers, one receipt that retires it on all of them.

The series, and where this fits

The bstack pattern series rebuilds the building blocks nearly every product needs so that the data defends itself: state proves itself on the BSV ledger, BSV Layered Multicast delivers each publication once to every subscriber, and readers verify instead of trusting whoever served them.

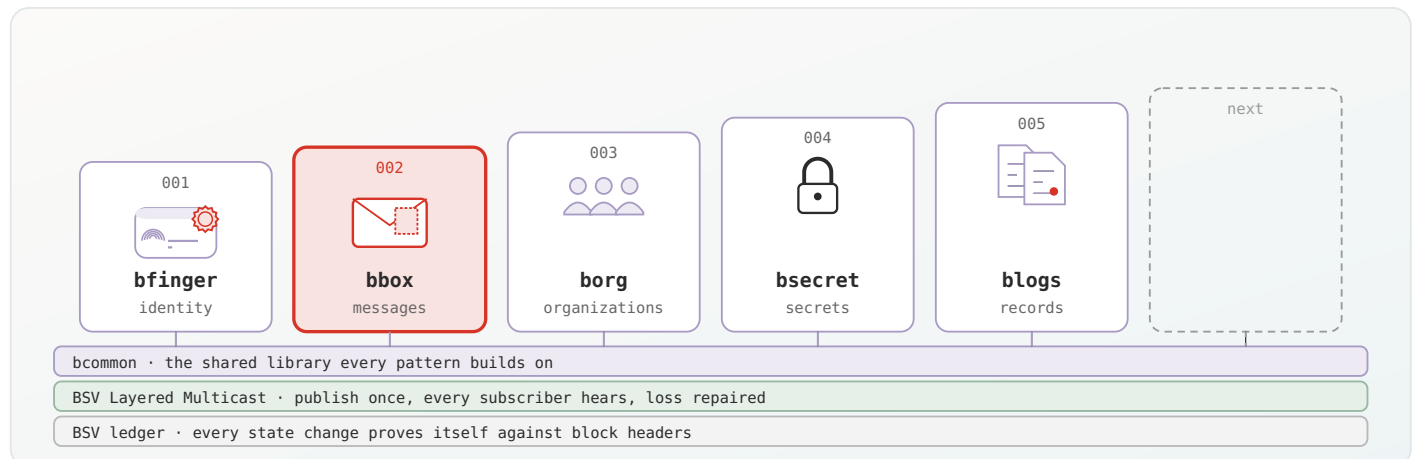


Figure 2: the series as one build. bbox, in red, is the second pattern.

So far. Pattern 001, **bfinger**, made a name and its record prove themselves from any host, and made every verified identity payable. It established the foundation the whole series reuses: a few hundred bytes mined per change, the data riding beside them in a transaction built never to be mined, and hosts as interchangeable replicas. It publishes to everyone.

This one. bbox is the first pattern that speaks to one party. It keeps everything bfinger built and adds privacy, postage and payment: the pieces any product needs to notify, invoice, or hand one party one thing. Where bfinger lets you pay a verified identity, bbox carries the payment and its note to them in one sealed envelope.

The problem

Every application eventually needs to put something in front of one person: a notification, an invoice, a payment notice, a key, a document. The tools for it, email, in-app inboxes, push services, webhook queues, all share three weaknesses. One operator holds the box, so the box is only as reliable and as honest as that operator. Nothing in the message proves who sent it or that it arrived unaltered, so spoofing and disputes are permanent costs. And sending is free, so every system grows a spam filter, a rate limiter and an abuse team.

Money makes it worse. A payment and the message about it travel separately, through different providers, and reconciling the two is somebody's job.

What bbox does

A recipient opens an **office**, a readable name with ten random letters appended so nobody else's office collides with it, and hosts carry it. Inside the office, **boxes** are labels a sender picks: **inbox**, **invoices**, **alerts**. A sender seals an **envelope** to the recipient's identity key, optionally with a payment inside, and publishes it once.

```
$ bbox send 03a1...77c2 -m 'a coffee' -pay 5000
$ bbox list
1 envelope(s) in the inbox of support_qzxkvbwmtr, 2 of 2 host(s) answering
$ bbox internalize 3f9a...c410
internalized 5000 sat from 02c6...9a1e: payment 77d2...e1a0 ...
```

Figure 3: send with a payment inside, list from every keeper, claim. Keys and names are illustrative.

The recipient lists the box from any keeper, or several, and checks every envelope against block headers it validated itself. A keeper that answers something that does not verify is named, and nothing it answered is shown: what the recipient sees comes only from keepers whose answers check. Keepers that answer different sets are reported one by one; the union is shown, never merged silently. One signed **receipt** then retires up to 64 envelopes on every keeper at once.

How it works

1. **Envelopes are never mined.** Each rides a transaction built so it can never be mined, spending one small output of the sender's mined **funding tree**. That binds it to the sender's key, lets anyone verify it, and lets the sender retract it with one sweep.
2. **Postage is structural.** One funding output carries one envelope, ever. No unspent output, no envelope: flooding a box costs the flooder real postage, so no central gatekeeper decides who may send.
3. **Content is sealed to the recipient.** Only the recipient's key opens it. Who wrote to whom, and when, is visible; what was written is not.
4. **Acknowledgement needs nobody's word.** A receipt is a signed record every keeper applies, so "read" means the same thing on every host, and hosts keep the receipt, free to ask for, as long as they keep what it acknowledged.
5. **The payment travels encrypted.** A payment inside an envelope is a signed transaction only the recipient can see, so no keeper and no subscriber can broadcast it first. It settles between the two parties when the recipient broadcasts it; no host sees, relays or settles it.

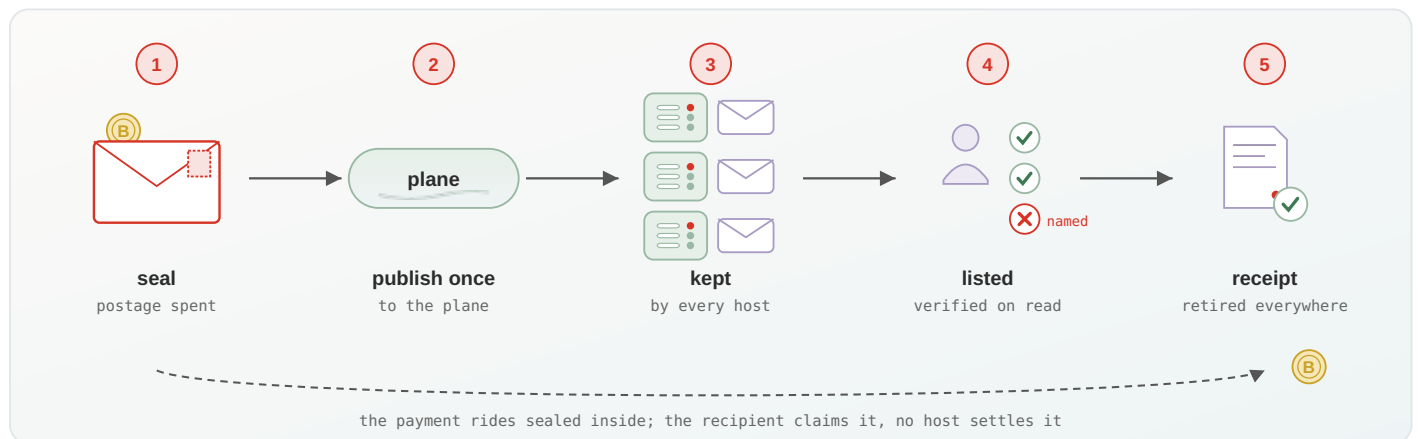


Figure 4: the life of an envelope.

Words in this brief. *Office*: a recipient's address on the network, carried by hosts. *Envelope*: one sealed message. *Funding tree*: one mined transaction with many small outputs; each envelope spends one, which is its postage. *Receipt*: a signed acknowledgement every host applies. *402*: the web's "payment required" answer, used here for priced questions.

Why it is new

- **The box has no single keeper.** Every host that carries an office holds the same verifiable envelopes. Losing a host loses nothing; a host that withholds shows up when its answer is compared with the others, and one that answers anything forged is named and ignored.

- **Sender authenticity is checked by every host**, at the level of the transaction itself, not by a header a server can forge.
- **Postage without a toll booth**. Spam costs postage, not permission: the cost is in the envelope's structure, so no central service has to grant or meter sending; filtering what arrives stays with the recipient.
- **Payment and message travel as one**. The notice and the money arrive together, and the payment cannot be front-run by anyone who relays the envelope.
- **Sender-side retraction**. A sender can take an envelope back with one mined sweep that every honest host honors: a fact on the ledger, not a request to one operator.
- **The free read is pinned by the specification**. Listing a box, reading a sender's envelopes and receipts are free on every conforming host, always; no priced question may be a superset of a free one.

Built to scale: BSV Layered Multicast

A box worth trusting has several keepers, ideally run by different parties. Point to point, every envelope is uploaded once per keeper, and a keeper that was down misses what was sent while it was away.

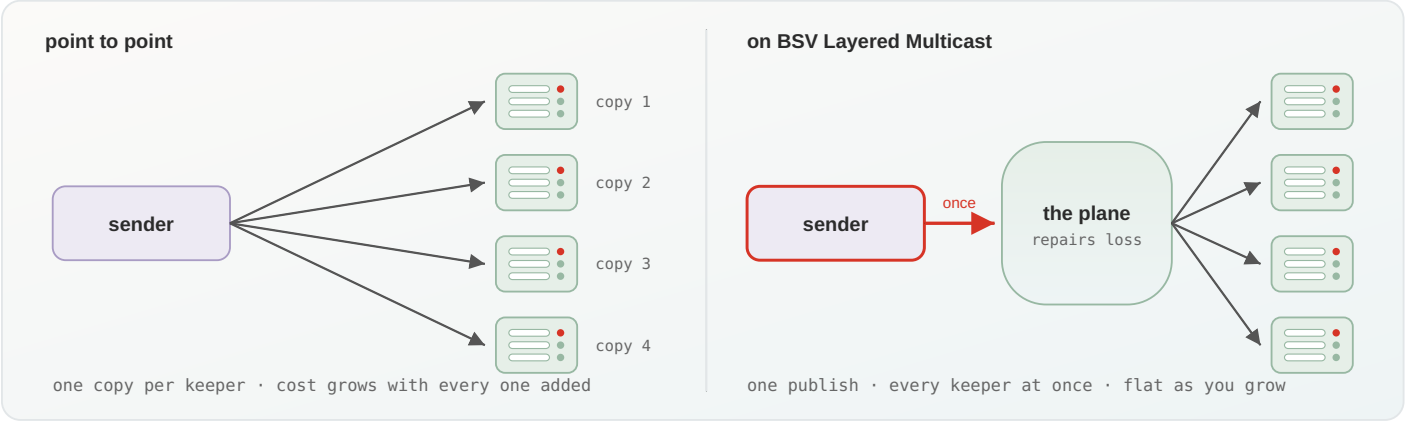


Figure 5: one publish per envelope, whatever the number of keepers.

On BSV Layered Multicast a sender submits once; every keeper subscribed to the office receives the envelope, and loss is repaired by the network, so a keeper that was briefly down catches up from the network rather than from a peer. A box is an index, not a chain, so there is no per-recipient bottleneck; busy recipients spread load across several offices by construction. bbox runs without the network too, sending to a handful of keepers directly; past a few, the plane is what removes the cost.

What it means for the business

Who	Pays	Gets
A sender	postage: a small share of a funding tree per envelope, in miner fees	proof of sending, retraction at will, a payment that arrives with its notice
A recipient	the same for a receipt; reading is free	several keepers, every envelope verified, lying keepers named
A host operator	its hosting and the delivery it elects	paid history and retention, settled through a 402 the client pays

- **Notifications, invoices and receipts with proof attached**. Disputes about whether something was sent, and by whom, are settled by the record.
- **Monetization without lock-in**. Hosts sell availability, latency and retention, never exclusivity; readers pay hosts, never the plane, and the delivery meter never bills a payment as a payment.
- **Abuse economics flip**. Postage makes bulk unsolicited sending cost money, which a filter never did.

Honest limits

- **Metadata is public**. Sender, recipient, box, times and sizes are visible to every host on the office and to anyone who follows its topic; content is not.
- **No forward secrecy**. A recipient key stolen later opens old mail.
- **Retraction and acknowledgement are not erasure**. Whoever already read or copied an envelope keeps it.
- **No push**. A recipient polls its keepers.
- **Messages are small**. Larger files travel by reference.
- **A payment must be claimed in time**. Until it is broadcast, the sender can still spend the coin elsewhere; an envelope's payment has an expiry.

What comes next

bbox speaks to one party. Pattern 003, **borg**, speaks for many: organizations and groups, with membership as certificates that can really be revoked, every change recorded in one order nobody can forge, and a group key that changes the moment anyone leaves. It is the layer team chat, secret grants and delegations stand on.

Run it on 1BSV

The 1BSV network is the commercial operation of BSV Layered Multicast, run by Lightweb Inc. For an application built on bbox:

- **Senders ride free.** Publishing into the network is never billed, and neither are the network's repair retransmissions.
- **Keepers pay for what is delivered to them**, and for nothing else: overlay objects with their proofs on an application account, which also carries the live header feed to verify against. No subscriptions, no minimums, no commitments.
- **Early access.** The overlay lane is onboarding its first users. Onboarding is self-service with one approval step.

Go deeper

- *The bstack: publish once, prove everything, and let users ride free* (Lightweb Inc., Paper 009): <https://1bsv.net/papers/bstack-patterns.pdf>
- *The overlay object plane* (Lightweb Inc., Paper 006): <https://1bsv.net/papers/overlay-object-plane.pdf>
- Pattern 001, bfinger: <https://1bsv.net/patterns/bfinger.pdf>
- The pattern series: <https://1bsv.net/patterns.html>

1BSV - BSV Layered Multicast - cash moves fast on 1BSV. · 1bsv.net

Lightweb Inc. · Jeff Harris · jeff@lightweb.net · github.com/lightwebinc

Unbounded Solutions for a Small World™ · © Lightweb Inc.