

borg: teams whose keys change when someone leaves

borg gives an organization a membership roll and its groups a shared key that changes whenever anyone leaves. Membership is a certificate that one mined spend revokes for every verifier; every change to a group lands on that group's own chain, in one order nobody can forge; and every host refuses a removal that does not start a new key. BSV Layered Multicast delivers each change once to every host that carries the organization.

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

Pattern 003 · 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.

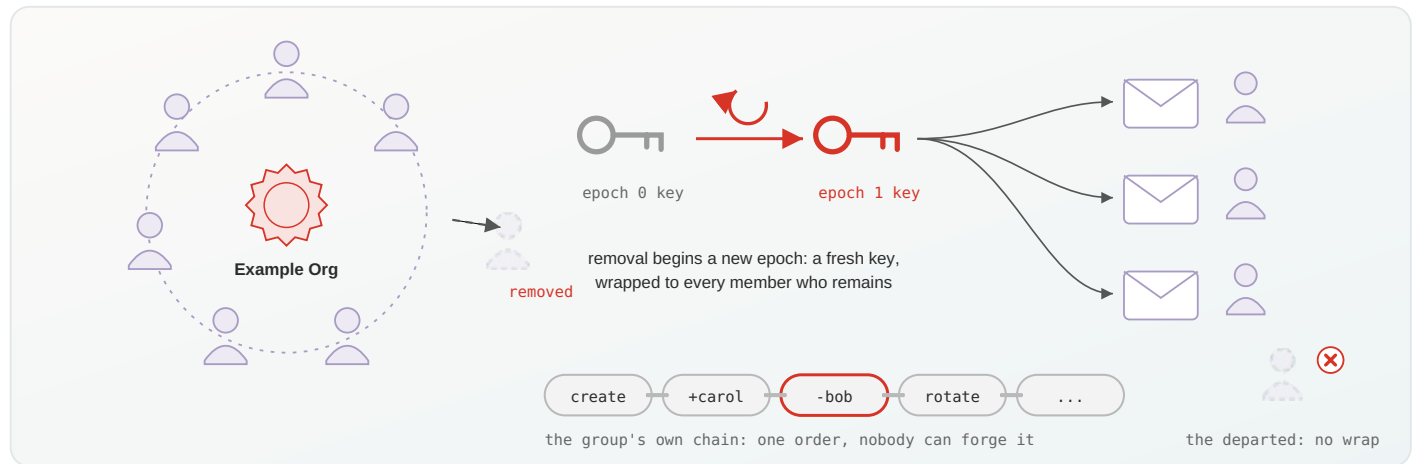


Figure 1: removal is a rekey by construction. The departed member is named by no wrap of the new key.

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.

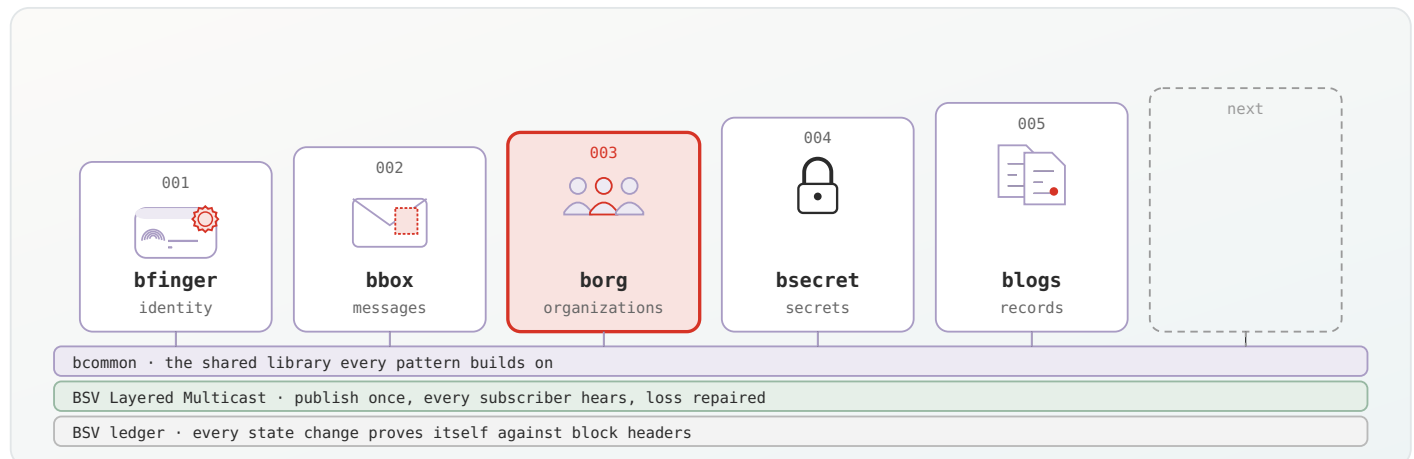


Figure 2: the series as one build. borg, in red, is the third pattern.

So far. bfinger (001) made an identity prove itself and made it payable. **bbox** (002) made a private, paid, many-keeper message to one party. Both are about one author.

This one. borg is about many. It gives the series organizations, groups and roles, and it closes a gap bfinger leaves open: today the domain names an identity's key, and borg is where handle certificates are verified. It is also the layer the next member stands on: secrets shared by a team, and anything a group must open and a former member must not.

The problem

Every multi-user product carries a directory: who belongs, in what role, to which groups. And most products handle the hard moment the same way: when someone leaves, their account is disabled, but the keys that protected shared material are not changed. In most products, removing a member is an access-control decision on the server, not a cryptographic one. The departed member's copy of the old key still opens whatever they can still reach.

The directory itself is one operator's database. Its order of events, who was added before whom, who was removed when, is whatever that database says, and two copies of it disagree the moment they are made.

What borg does

An organization creates groups, issues members certificates with a role and an expiry, and adds them to groups. Each group has a key, one generation of which is called an **epoch**. borg delivers the key; the application encrypts its own content under it.

```
$ borg org create 'Example' -member <alice> -member <bob>
$ borg group create engineering ...
$ borg org remove <bob> -group engineering      # a new epoch
epoch 1 CANNOT OPEN: removed at seq 1 ... no wrap of a later epoch names this key
```

Figure 3: create, remove, and what the removed member sees. Keys are placeholders.

- **Offboarding** removes a member from every group and revokes every certificate in one step.
- **A role change** is a revocation and a fresh certificate; the member's groups do not change, because a group names keys, not roles.
- **A suspected leak** is answered with a rotation on demand, without any change of membership.

How it works

1. **Membership is a certificate with a real off switch.** Each member holds a certificate (BRC-52) issued by the organization's key, with a role, a scope and an expiry. Its revocation point is the very output that funds it, so one mined spend revokes it for every verifier and retracts it from every host at once.
2. **Every group change lands on the group's own chain.** Each change spends the group's previous state. Two changes made at once spend the same output and only one is ever mined: total order with no coordinator, no merge, no server deciding who won.
3. **Removal is a rekey.** The rule every host checks: no removal is admitted without a new epoch, and no new epoch is admitted that reuses the old one. The new key is drawn at random, committed to before anyone holds it, and wrapped (encrypted) separately to each remaining member. No wrap names the departed.
4. **Readers check everything against their own block headers,** and fail closed: a member list must match the commitment on the chain, and hosts that answer differently are reported, never silently merged.

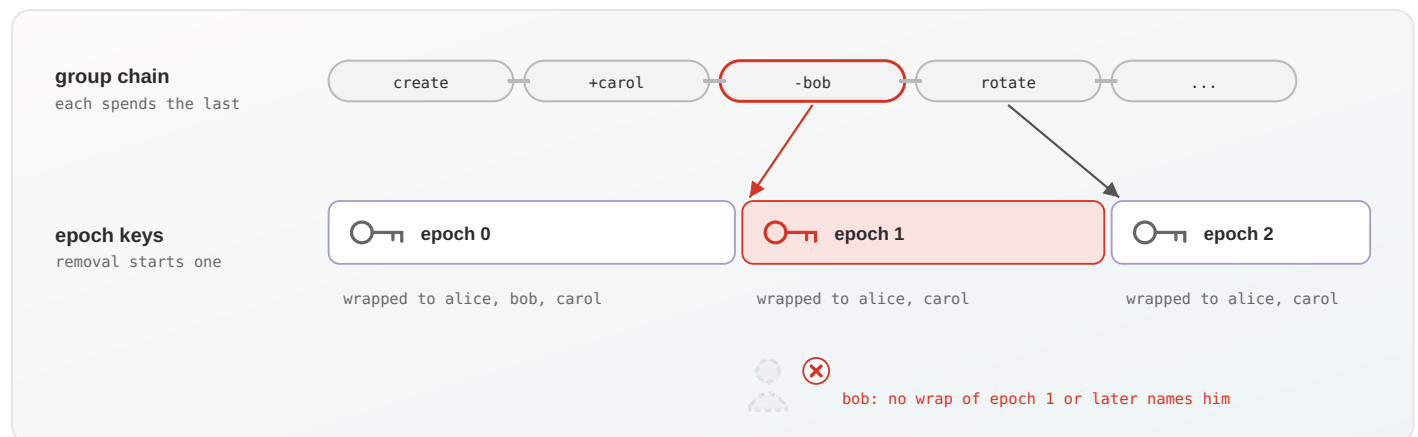


Figure 4: each change spends the last; each removal begins an epoch whose key the departed never receives.

Words in this brief. *Certificate*: a signed statement that a key holds a role, until it expires or is revoked. *Epoch*: one generation of a group's key. *Wrap*: the group key encrypted to one member. *Chain*: a sequence of mined records, each spending the one before, so their order cannot be disputed.

Why it is new

- **Rotation on removal is enforced by every host**, not by the good intentions of whoever runs the admin console. A removal without a rekey is simply not accepted.
- **Sequence without a server.** Group messaging standards leave the sequencing of changes to a delivery service. Here the group's own chain settles which change applies at each step, and nobody operates it.

- ▶ **One spend does two jobs.** The revocation point is the certificate's funding output, so a single sweep revokes for every verifier and retracts from every host.
- ▶ **A bounded removal window.** A member writing to a group must have checked recently, through a spend index rather than a host, that the group state it writes under is still current; a host that withholds a removal cannot extend the window.

Built to scale: BSV Layered Multicast

A team directory is read constantly and changed rarely, by every service and device the team uses. Point to point, every change and every wrap is uploaded once per host. Removing one person from ten groups is ten changes, each uploaded to every host in turn.



Figure 5: one publish per certificate or wrap, whatever the number of hosts.

On BSV Layered Multicast every certificate and every wrap, the bulk of what a change publishes, is submitted once and reaches every host carrying the organization, repaired by the network; the small mined record of the change is also handed to each host the organization names. The organization and each group are separate chains, so changes to different groups move in the same block; removing one person from ten groups is ten changes that can all be mined together. borg runs without the network too, sending to a few hosts directly; past a handful, the plane is what removes the cost.

What it means for the business

Who	Pays	Gets
The organization	miner fees per change, a fraction of a cent; certificates and wraps ride never-mined transactions, funded in batches by one mined tree	a roll and key schedule nobody can quietly alter
A member or reader	nothing to check membership, a group, an epoch key or a revocation	proof of who belongs, verified on its own machine
A host operator	its hosting and the delivery it elects	a seam to sell retention of ended epochs and bulk history

- ▶ **Offboarding that actually closes the door.** The departed member is cut out of every future key by construction, and the record shows when.
- ▶ **Auditable membership.** Who was in which group, from when to when, is a chain any auditor verifies against headers alone.
- ▶ **A substrate, not a silo.** Secret vaults already reuse the same roll and the same epochs, and chat rooms and delegations are designed to, instead of each inventing a directory.

Honest limits

- ▶ **Membership is public metadata.** Who belongs to which group, and when they joined and left, is visible; an opaque setting hides roles and scopes, not membership.
- ▶ **Rotation protects the future, not the past.** A removed member keeps what they could already read; nothing can make old content unreadable to them. There is no forward secrecy against a compromised key.
- ▶ **Rekeying is linear.** Removing one member costs one wrap per remaining member, so very large groups wait for a tree-based construction.
- ▶ **Removal takes effect when mined and seen,** a window of about a block, plus the propagation delay and each writer's freshness bound.
- ▶ **The organization's key is everything.** A compromised organization key means a new organization.

What comes next

borg gives a team its membership and a key that follows it. Pattern 004, **bsecret**, is the first application built on top: a secrets vault whose hosts hold only ciphertext, where adding a reader is adding a group member, removal forces a re-key, and the whole trail of grants, versions and rotations is a set of mined facts an auditor checks alone.

Run it on 1BSV

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

- **Organizations ride free.** Publishing into the network is never billed, and neither are the network's repair retransmissions.
- **Hosts 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>
- Patterns 001 bfinger and 002 bbox: <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.