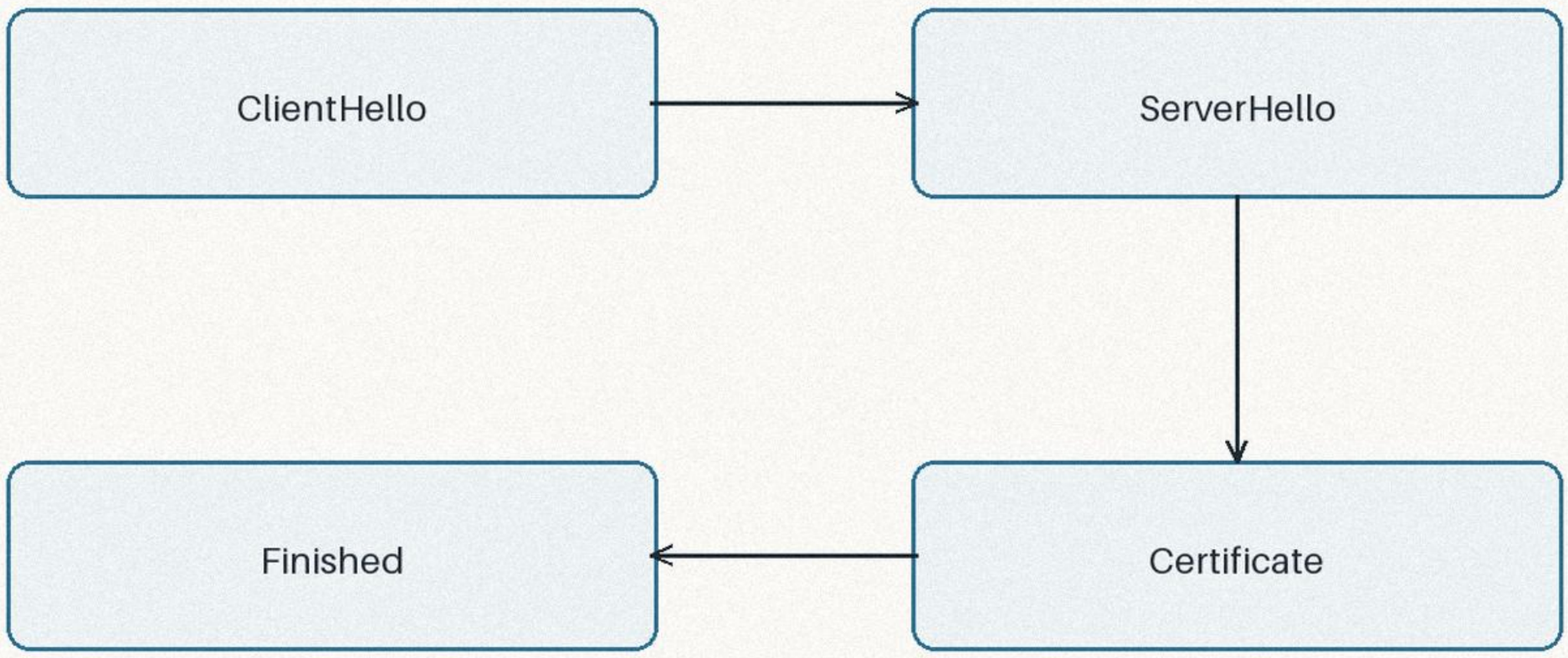
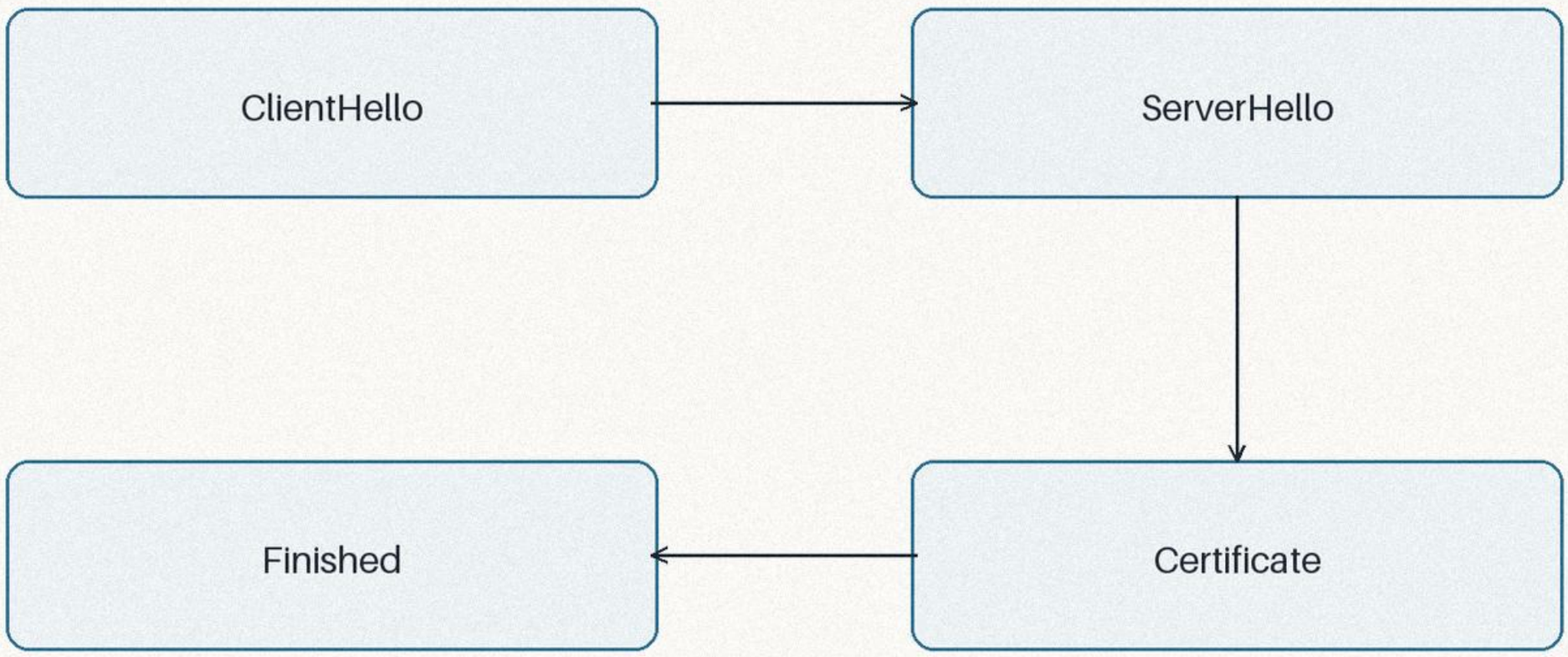
Small Systems Notebook



Working notes on running small web services without drama.

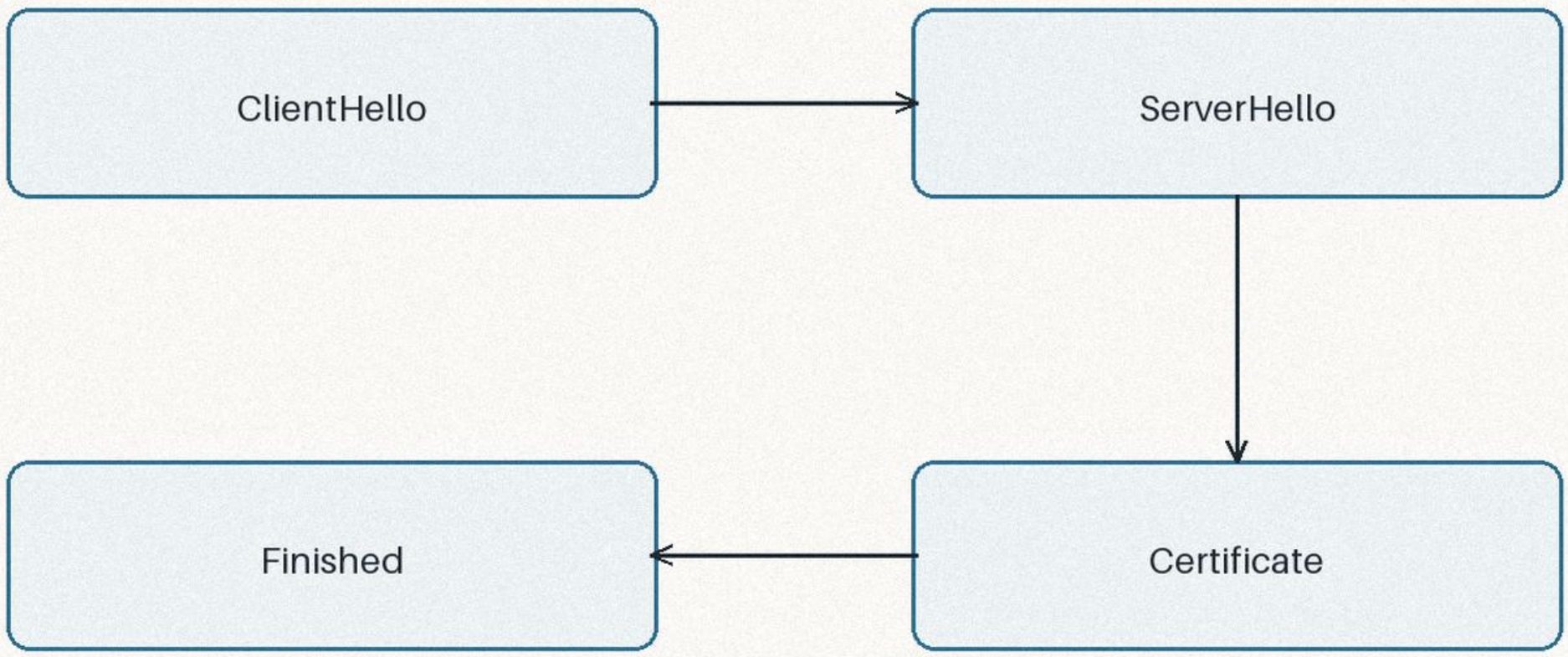
Printed from cedar.sanriodevgroup.org

Reading a handshake trace



What each message in a TLS 1.3 handshake is for, in plain words.

Reading a handshake trace



The client hello carries the name of the site and the key shares; the server answers with its share, then everything after that is encrypted. The certificate travels inside the encrypted part, which is why a passive observer sees the name but not the certificate.

Session tickets arrive after the handshake has finished. They are small, but their size depends on the server software, so two servers configured differently can be told apart by them alone.

Choosing DNS TTLs



Five minutes before a move, a day when nothing is planned.

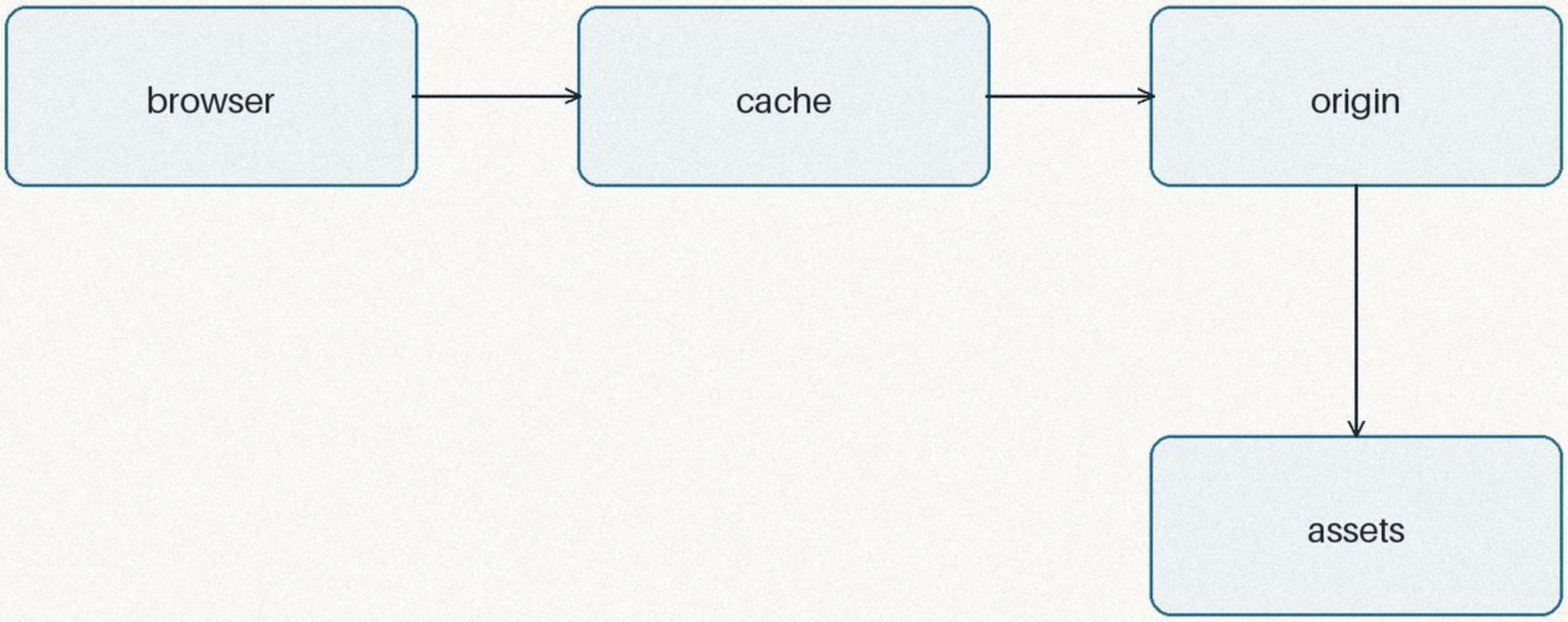
Choosing DNS TTLs



A short TTL costs a little latency and a few more queries; a long one costs time when an address has to change. The practical compromise is to lower the TTL a day before planned work, do the move, and raise it again afterwards.

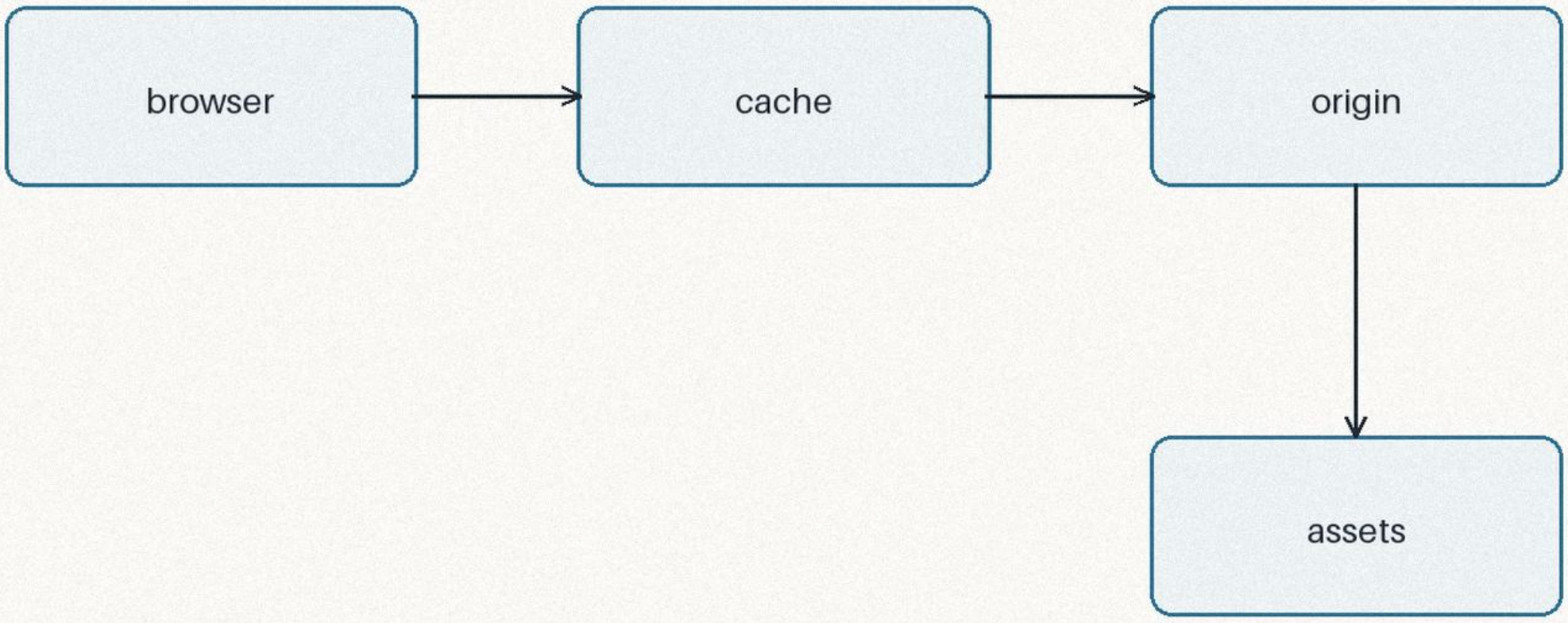
Negative answers are cached too, for the SOA minimum. A record created a minute after someone asked for it may stay invisible to that resolver for a while, which looks like a broken record but is only patience.

Cache headers that age well



Long-lived assets, short-lived pages, and the one header that makes both safe.

Cache headers that age well

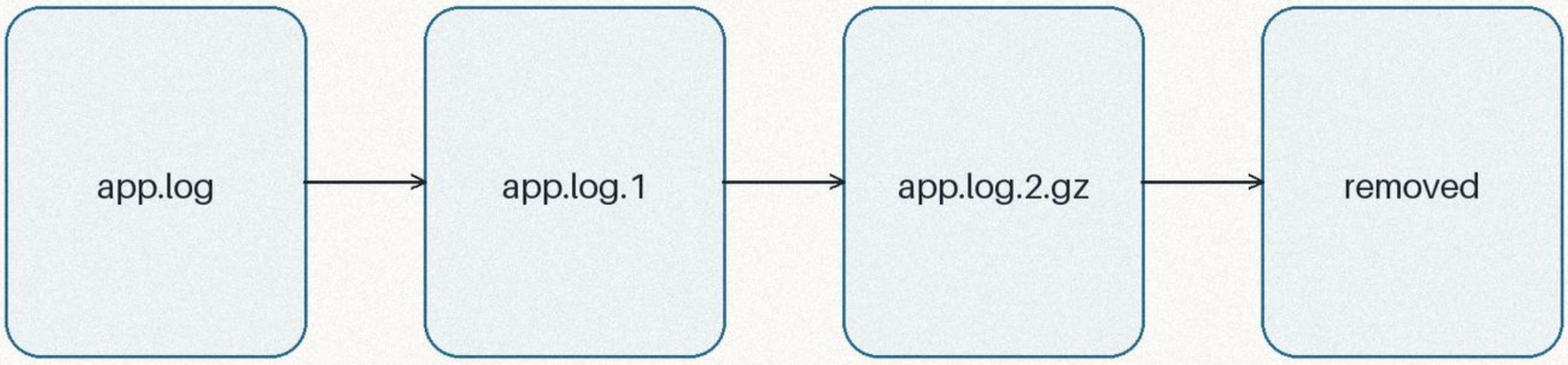


The simplest cache policy that has survived several redesigns: fingerprinted assets get a year and immutable, HTML gets max-age=0, must-revalidate. Browsers then ask for pages on every visit but almost never for stylesheets or images.

The part that is easy to get wrong is the fingerprint itself. It has to change when the bytes change and only then, so it should be computed from content, not from a build number. A build that produces identical files must produce identical names.

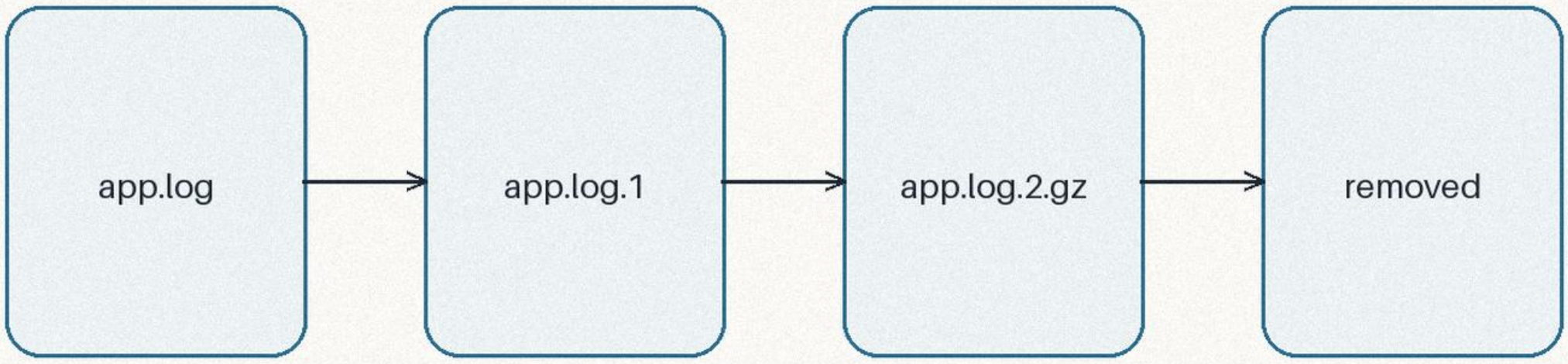
/assets/site-3f9a1c2e.css Cache-Control: public, max-age=31536000, immutable
/notes/ Cache-Control: public, max-age=0, must-revalidate

Log rotation without surprises



Size limits, copytruncate, and why the daemon should be told.

Log rotation without surprises



Rotation by size catches the day when a client misbehaves and writes a gigabyte of errors in an hour. Rotation by time alone does not. A cap of a few files of a few megabytes each is plenty for a small service.

copytruncate is convenient but loses the lines written between the copy and the truncate. When the program can reopen its log on a signal, prefer that and send the signal from the rotation hook.