History & Overview

J.D. Falk, Return Path Inc.

Mail Abuse Reporting Format (MARF) Working Group

IETF 77—Anaheim, California, USA

- Complaint Feedback Loops
  - Rationale
  - History
- ARF ➠ MARF
  - History
  - Installed Base
  - Format Overview

Complaint Feedback Loops

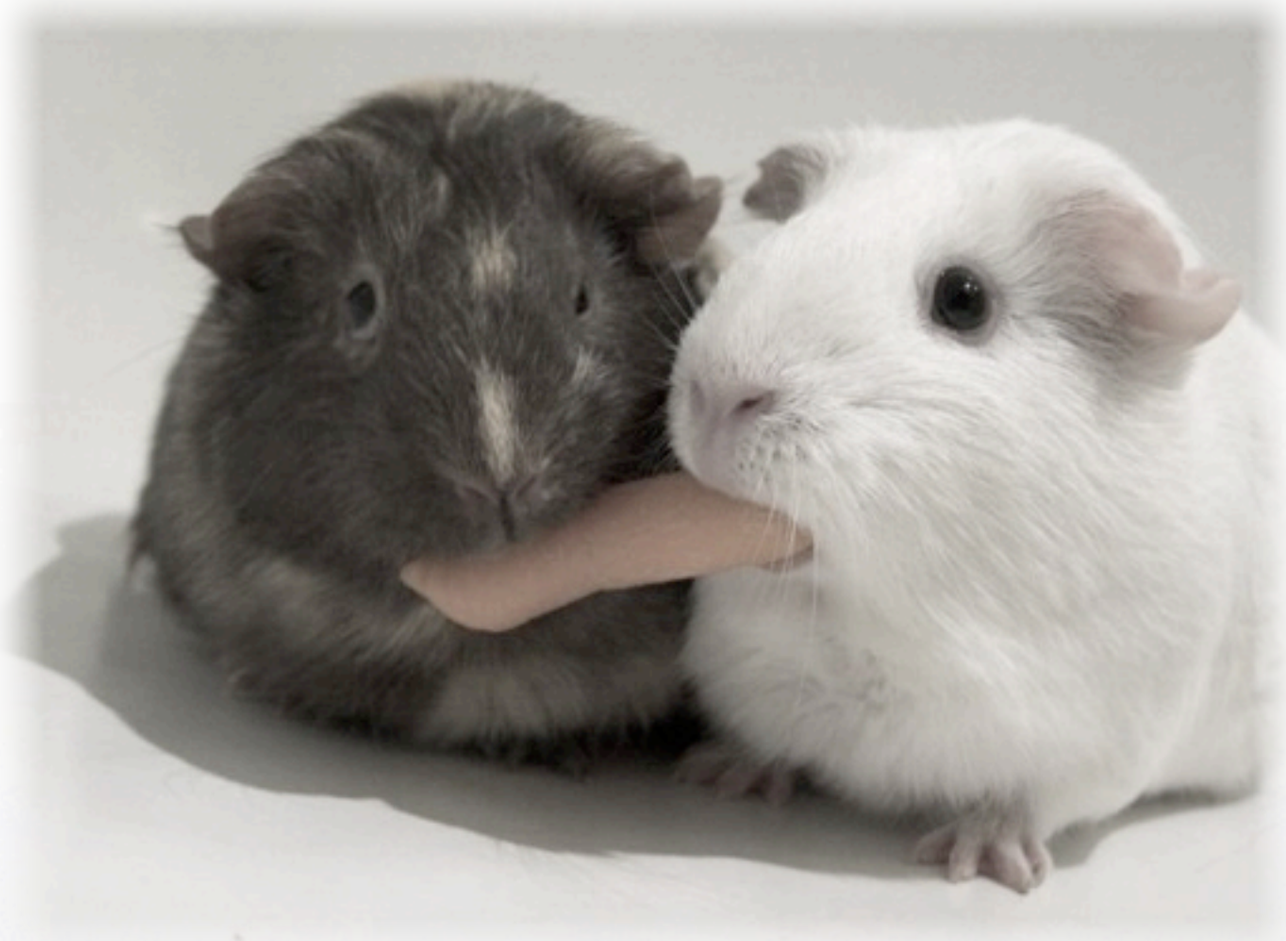
end users receive spam

they want to complain

complaint mechanism

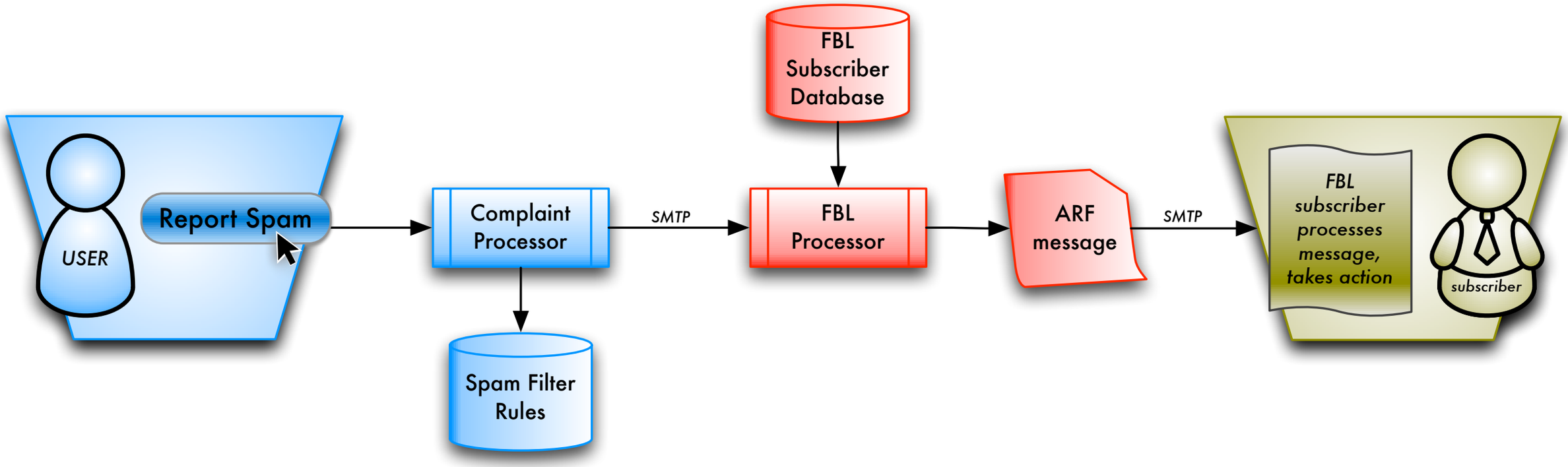
- users click a “spam” button in their mail client (MUA)
- the message is returned to their mailbox provider *via an out-of-scope process*
- the mailbox provider collects complaints, and processes them *using their own out-of-scope logic*

usually to improve their spam filters
but spam filters don’t address the root cause, the source of the problem



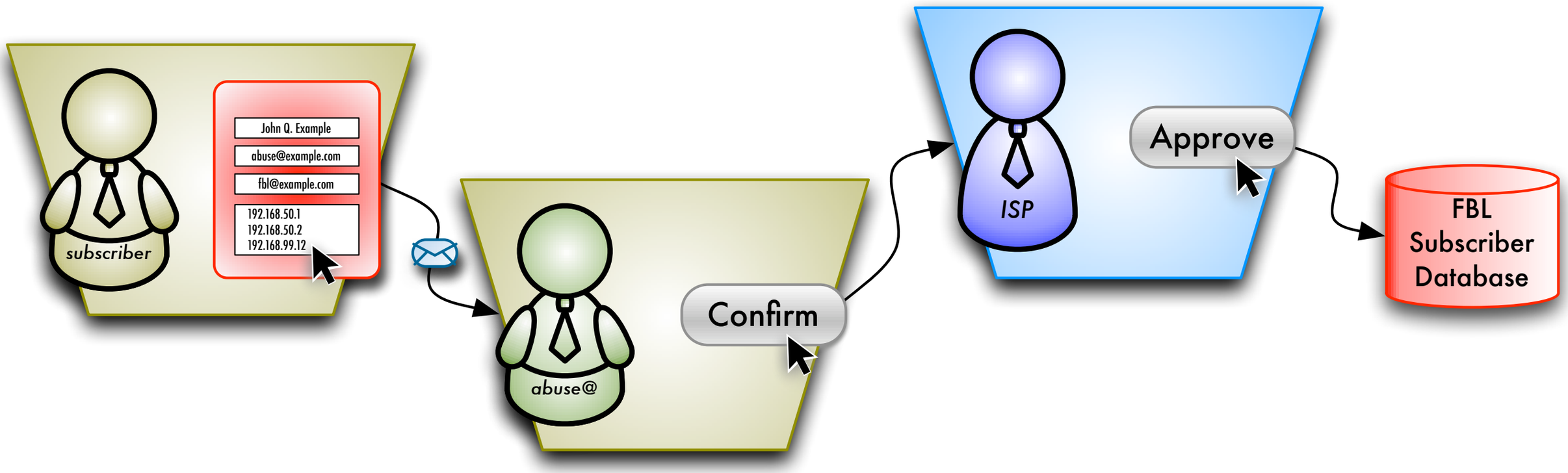
let's share!

so a few mailbox providers started forwarding complaints to each other


FBL
Subscriber
Database
USER
Report Spam
Complaint
Processor
SMTP
FBL
Processor
ARF
message
SMTP
FBL
subscriber
processes
message,
takes action
subscriber
Spam Filter
Rules

Complaint Feedback Loop

- **generators**
  - mailbox providers
  - a 3rd party working on behalf of the mailbox provider
- **consumers**
  - hosting companies
  - ESPs
  - direct senders (author or operator)
  - a 3rd party working on behalf of the author or operator
  - researchers



in nearly every case, the feedback flows from generator to consumer, or subscriber, by prior arrangement: the consumer requests feedback, and the generator decides whether to approve that request.

in this example, the consumer's abuse@ address is asked to confirm that this particular applicant is appropriately asking on behalf of that domain — a fairly common practice. however, the approval process is out of scope.

What You Got

- one new SMTP message per complaint (just like forwarding)
- headers & body dumped into body portion of a new message (forwarding inline)
- various encoding
- various whitespace
- *sometimes* explanatory text at the top

sometimes message/rfc822

ARF

ARF Timeline

- "spam" button appeared around 1998
- abuse reporting group formed in 2005
  - spinoff from MAAWG, plus a few experts
  - initially a closed discussion
- *feedback-report-00 — March 2005*

general recollection is that AOL had the button first, and later it became standard in webmail clients; there was also Vipul's Razor, a collaborative spam reporting system which became part of Cloudmark, who distribute a plugin for desktop MUAs

ARF Timeline

- larger, open discussion started
- first implementations began to appear
- *feedback-report-02 — May 2007*
- already the *de facto* standard

we weren't working very fast anymore, because a de facto standard was sufficient

ARF Timeline

- *feedback-report-04 — March 2008*
  - added DKIM failure reporting
- *feedback-report-08 — October 2009*
  - last non-IETF version

MARF Timeline

- *draft-ietf-marf-base — January 2010*
  - removed all report types not currently in use (they can come back as extensions)
- 1st MARF WG meeting

Installed Base: ARF Generators

- AOL
- BlueTie
- Comcast
- Cox
- Earthlink
- Microsoft
- RackSpace
- Outblaze
- Road Runner
- Tucows
- USA.net
- Yahoo!

(12 biggest ARF generators by volume, in alphabetical order)
just over half of these are operated by a 3rd party, and thus are on the same codebase
I know of about a dozen more implementations in progress

Known Outliers

- ARF reports generated by an MUA
- ARF reports sent to role accounts such as abuse@ without prior arrangement *(not recommended unless you are John Levine)*
- ARF reports generated from spamtrap messages, rather than complaints

these aren't necessarily out of scope, but they aren't the primary use case

MARF

draft-ietf-marf-base

- defines a new MIME type: **message/feedback-report**
- *Determination of where these reports should be sent, how trust among report generators and report recipients is established, and reports related to more than one message are outside the scope of this document.*

Stated Requirements

- both human and machine readable
- entire original email message enclosed
- machine-readable meta-data
- must be extensible

Unstated Requirements

- Must remain compatible with current installed base of generators & consumers
- Intended for software-to-software and software-to-human communication, rather than human-to-software or human-to-human
- Redaction (removing part of a message due to privacy concerns) *is* going to happen

thus, while primarily machine-readable, it should also be at least as human-readable as standard email headers

MIME parts

- multipart/report
  report-type: feedback-report
  - text/plain
  - message/feedback-report
  - message/rfc822 *or*
    message/rfc822-headers

text/plain portion

- an entirely human-readable section, often containing canned boilerplate

message/feedback-report

- various header-like fields contain metadata
- *Note that these fields represent information that the ~~receiver~~ report generator is asserting about the report in question, but are not necessarily verifiable. Report receivers MUST NOT assume that these assertions are always accurate.*

message/feedback-report
required fields

- Feedback-Type:
  - abuse
  - fraud
  - other
  - virus
- User-Agent:
  - *information only; no registry*
- Version: 0.1

there used to be other feedback-types, but they weren't used
I believe the DKIM failure report type will be reintroduced as an extension,
not aware of anyone intending to reintroduce the others

message/feedback-report
optional fields

appearing once

- Original-Envelope-ID:
- Original-Mail-From:
- Arrival-Date:
- Reporting-MTA:
- Source-IP:
- Incidents:

appearing multiply

- Authentication-Results:
- Original-Rcpt-To:
- Reported-Domain:*
- Reported-URI:

IETF®