Clustered Logging

with mod_log_spread

Theo Schlossnagle

<jesus@omniti.com>

The Speaker

Theo Schlossnagle — Pricipal @ OmniTI

- open-source developer
  - mod_backhand
  - Wackamole
  - daiquiri
  - OpenSSH/SecurID
  - Spread
  - etc.
- closed-source developer
  - Ecelerity MTA
  - Ecelerity Clustering

Agenda

- Understanding the Problem Space
- A Survey of Technologies
- Implementing Clustered Logging
- Understanding New Possibilities

Understanding the Problem Space

The Purpose of Logging

- Journalling the fact that a transaction has taken place.
- Correlating a series of transactions into a session.
- An audit trail.
- Forensics.
- Activity analysis to understand current trends and predict the future.

Basic Expectations

Logs are reliable.

Events are logged in the order they occur.

They can be partitioned by date.

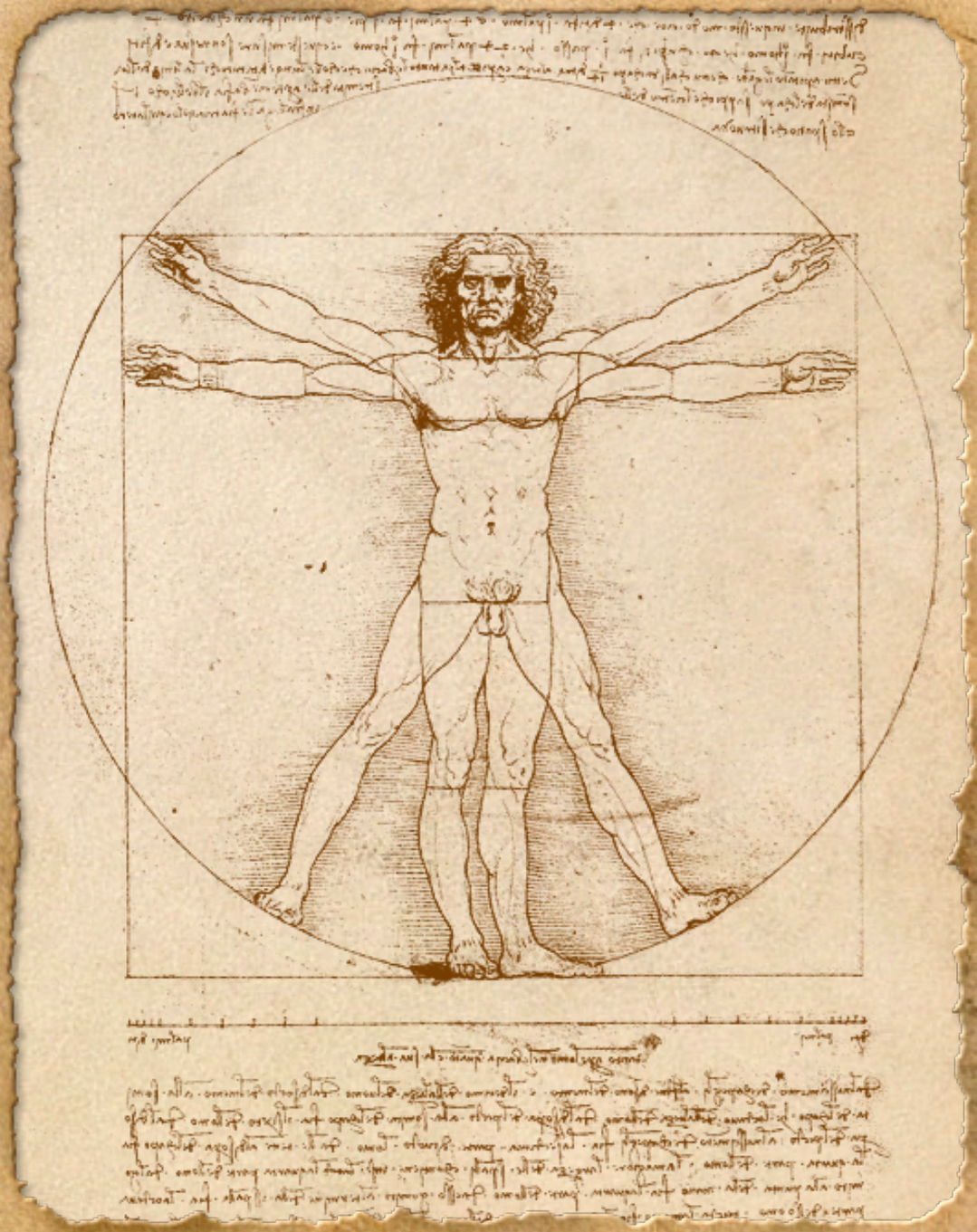
They can be multiplexed and demultiplexed on demand.

Introducing Clustering

- Clustering:

  several machines acting together to provide a single service

- Sessions may now be composed of a series transactions that occur on different machines.
- Ordering is "harder" and more important.



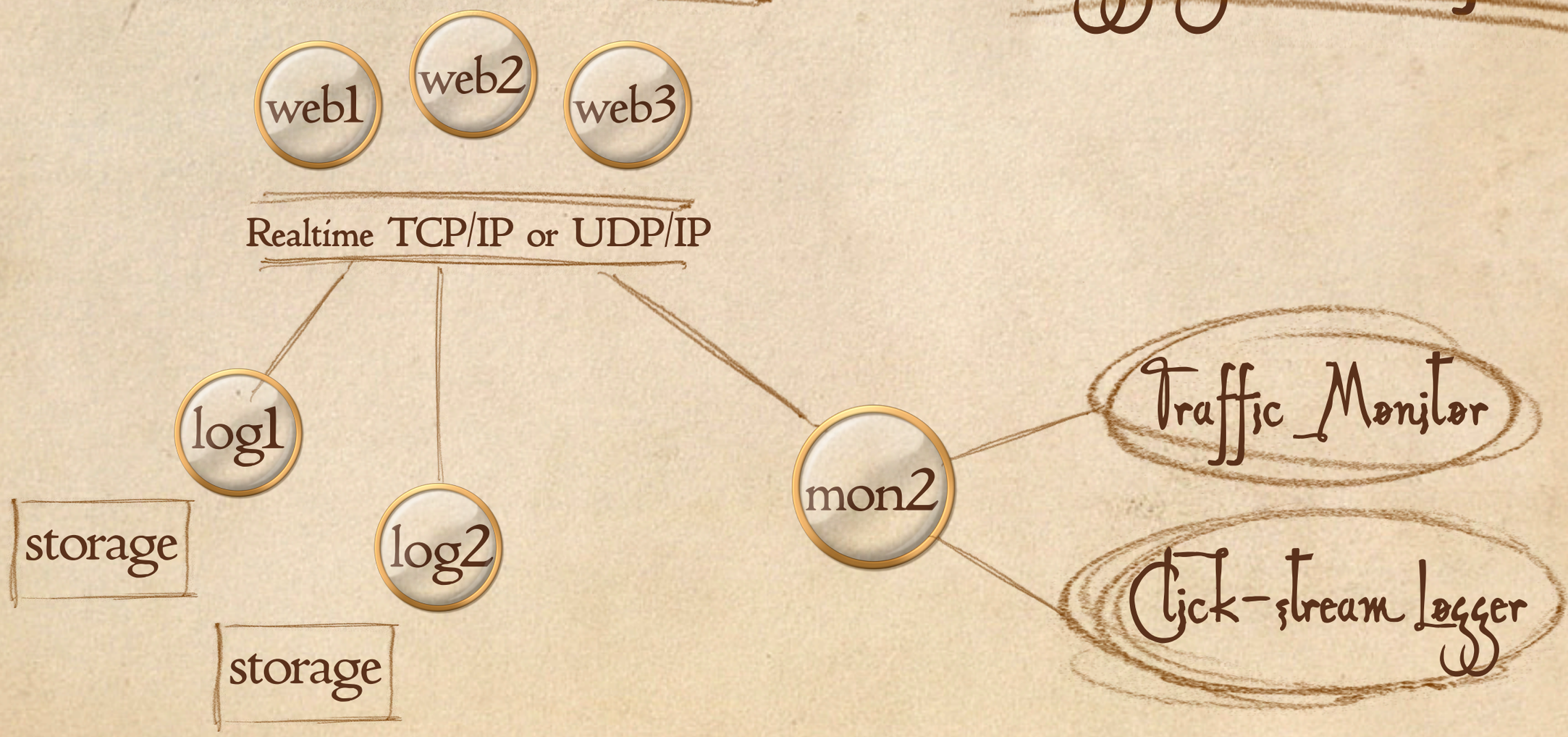
A Survey of Technologies

Traditional Logging

- Logs written locally on web servers
  - space must be allocated
- Consolidation happens periodically
  - crashes will result in missing data
  - aggregators must preserve chronology
  - real-time metrics cannot be calculated
- Monitors must run against log servers
  - monitors must tail log files
  - requires resources on the log servers

Traditional Approach

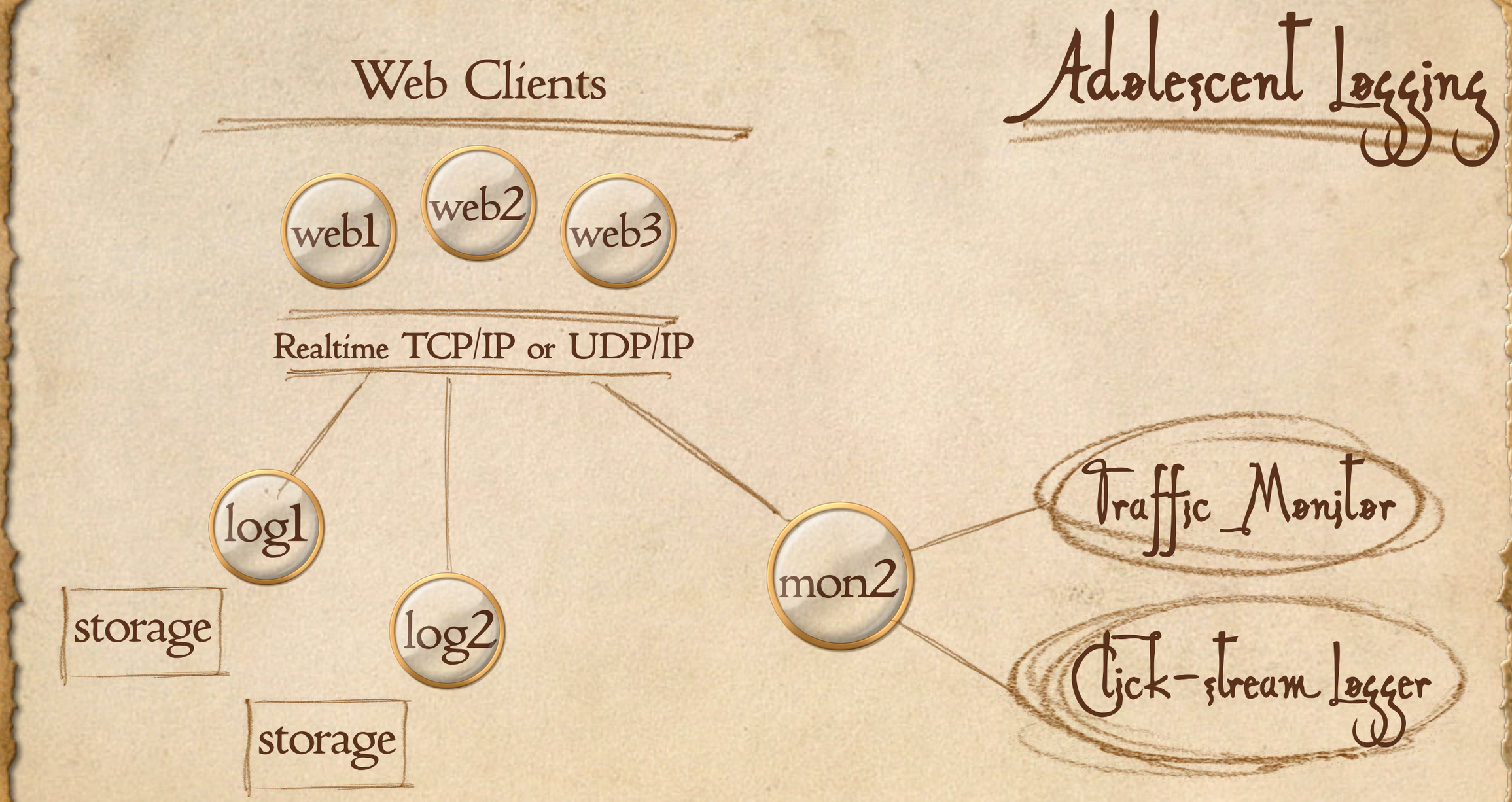
Logging in its infancy



Active Network Logging

- Logs written directly to log servers
  - UDP is unreliable and thus not useful
  - TCP is a point-to-point protocol
    - Two log server mean double traffic
    - Add a monitor and that's triple!
- Real-time metrics are possible
  - monitors must tail log files still (or publishers must send directly to the monitors... yuck!)

Network Approach

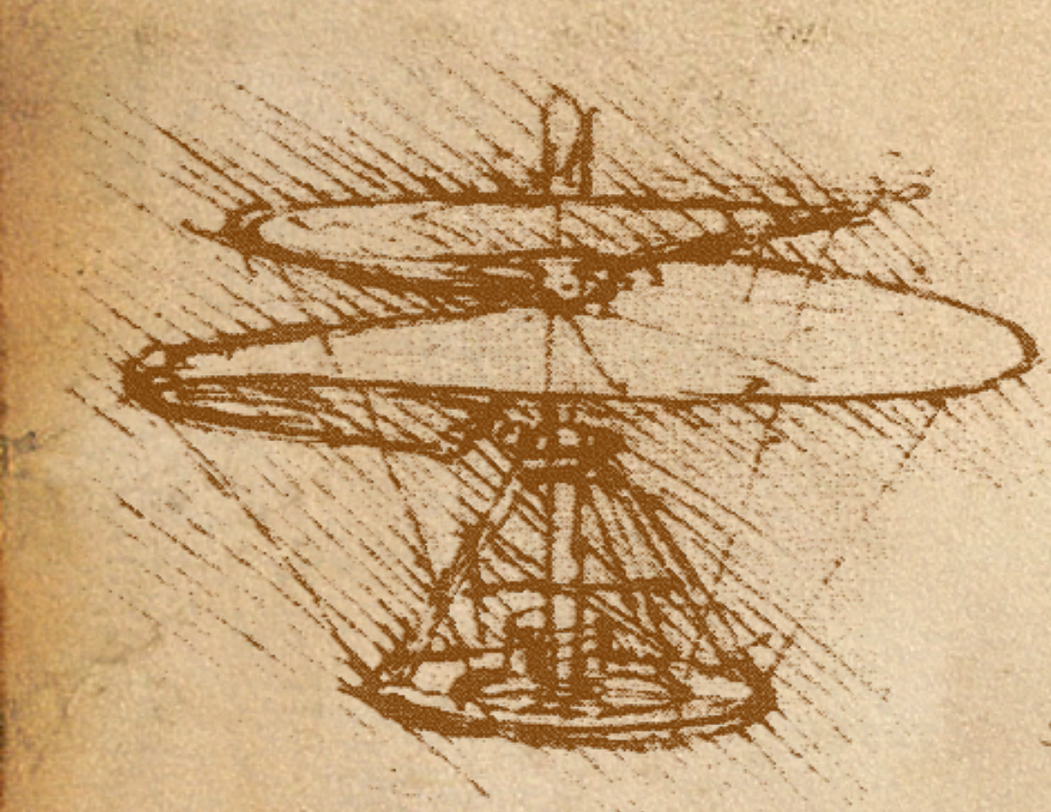


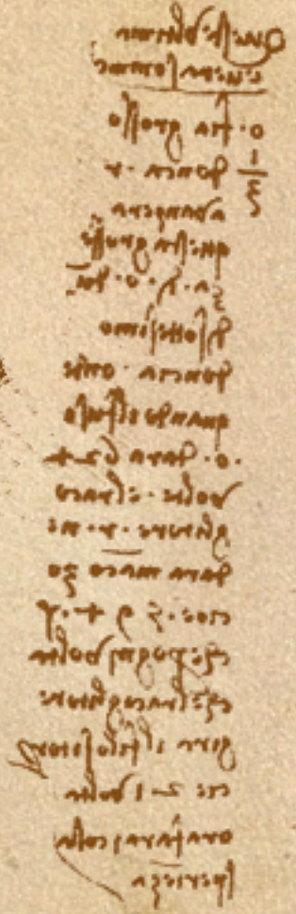
Passive Network Logging

- Logs constructed from sniffed traffic
  - The players no longer matter
  - Web servers can be added easily

- Drops logs!
  - When tested head-to-head with active logging frameworks we see loss
  - Missing logs is unacceptable

Passive Logging



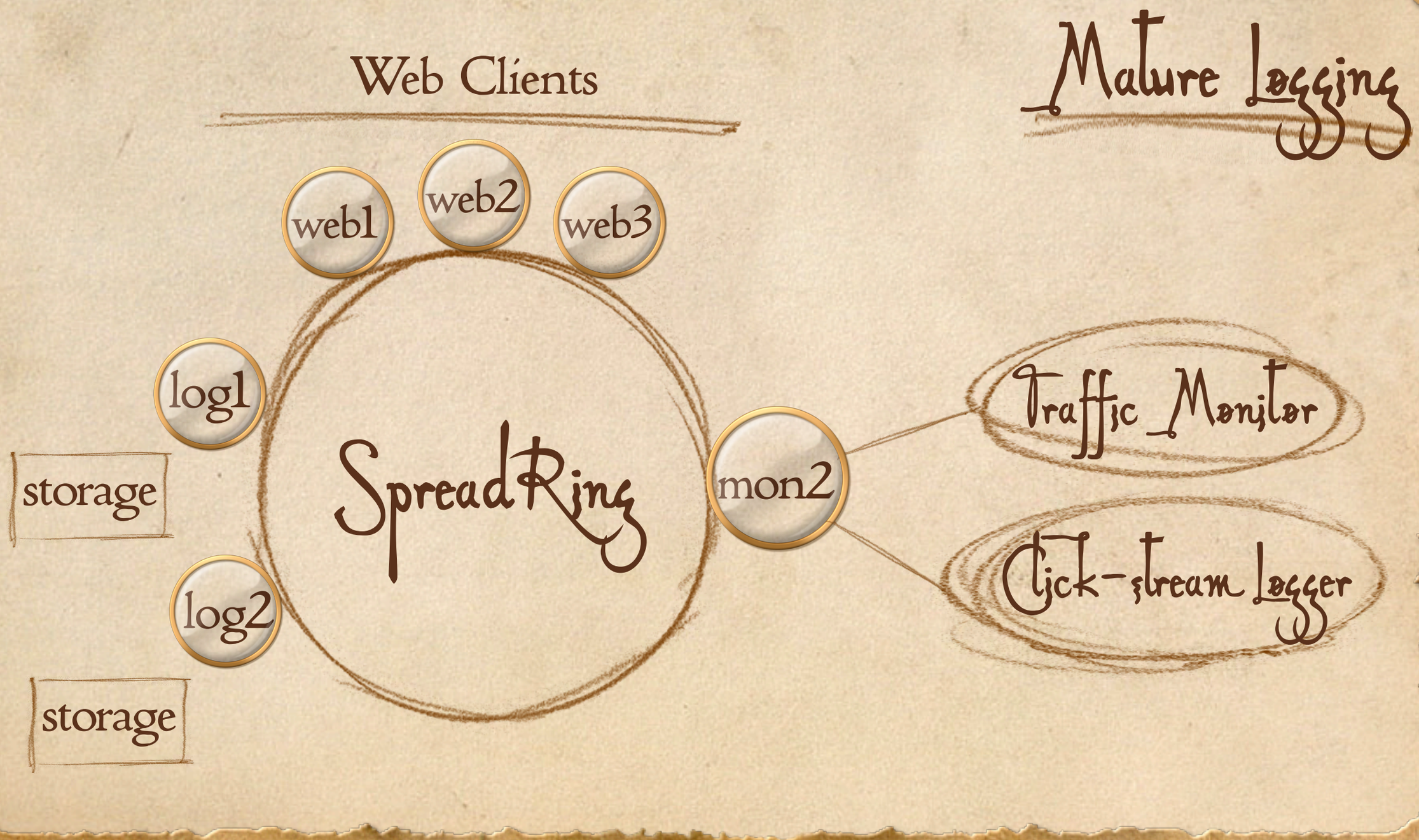


A lapse in judgement

mod_log_spread Logging

- Logs are published over Spread
  - Efficient reliable network multicast
  - Preserves global ordering of logs
- Multiple subscribers at no cost
  - well... almost zero
- Extends well beyond Apache
  - All logging (enterprise wide) can be utilize this publish/subscribe messaging bus

mod_log_spread



Clustered Logs Provide

- instant aggregation
- ordering
- publish/subscribe model
- multiple subscribers
- multiple subscribers
- multiple subscribers...

Multiple Subscriber Magic

- Data "feeds"
- Write them to disk
- Real-time analysis:
  - popular pages
  - concurrent sessions
- Who's online?
- Understand load-balanced click streams

Implementing Clustered Logging

So show me!

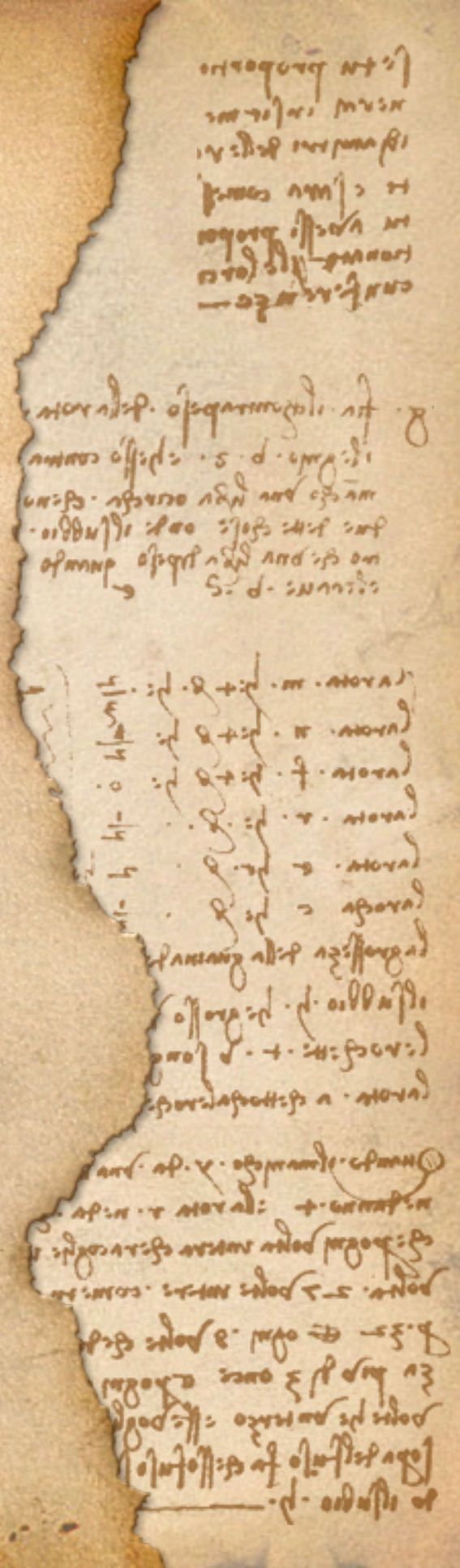
- Spread
- Apache 1.3 or 2.0
- mod_log_spread
- spreadlogd
- A spread client API for your favorite language:
  - Perl, Python, C Java, Ruby, PHP, etc.

Install Spread

http://www.spread.org/

A simple /etc/spread.conf:

```
DebugFlags = { EXIT CONFIGURATION }

EventLogFile = /var/log/spread/mainlog
EventTimeStamp

Spread_Segment  10.225.209.255:4913 {       # order matters
        admin-va-1      10.225.209.68       # staging server
        www-va-1        10.225.209.71
        www-va-2        10.225.209.72
        www-va-3        10.225.209.73
        samwise         10.225.209.240      # logging machines
        gollum          10.225.209.241      # monitoring machine
}
```



Install mod_log_spread

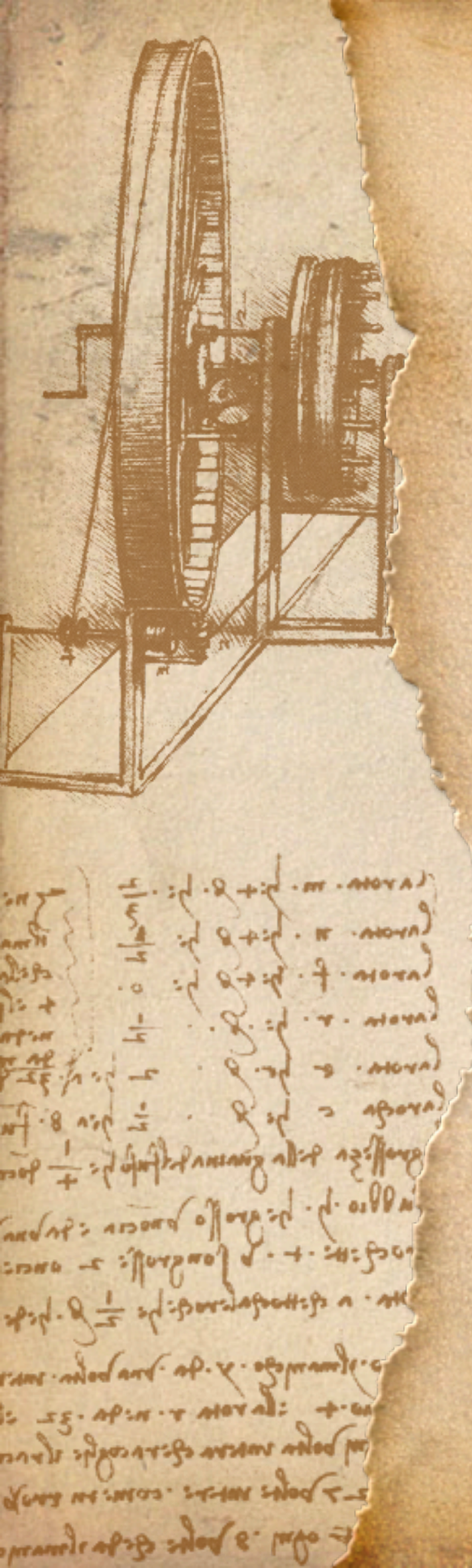
http://www.backhand.org/

A simple httpd.conf:

```
LoadModule log_spread_module  libexec/mod_log_spread.so
AddModule mod_log_spread.c
#AddModule mod_log_config.c
SpreadDaemon 4913

LogFormat "%h %l %u %t \"%r\" %>s %b" common

<VirtualHost coolsiteip:80>
        CustomLog $coolsite common
</VirtualHost>

<VirtualHost slicksiteip:80>
        CustomLog $slicksite common
</VirtualHost>
```

Verify it is working

```
; /opt/spread/bin/spuser -s 4913
User: connected to 4913 with private group #user#admin-va-1
User> j coolsite
==========================
Received REGULAR membership for group coolsite with 2 members, where I am member 1:
        #user#admin-va-1
grp id is 182571332 1092928408 2
Due to the JOIN of #user#admin-va-1
User>
==========================
received RELIABLE message from #ap25454#admin-va-1, of type 1, (endian 0) to 1 groups
(182 bytes): 68.55.183.91 - - [30/Oct/2004:11:48:51 -0400] "GET /~jesus/ HTTP/1.1" 200 57940 "-"
"Mozilla/5.0 (Macintosh; U; PPC Mac OS X; en) AppleWebKit/125.5 (KHTML, like Gecko) Safari/125.9"
```



Install spreadlogd

http://www.backhand.org/mod_log_spread/

A simple /etc/spreadlogd.conf:

```
BufferSize = 65536
Spread {
  Port = 4913
  Log {
    RewriteTimestamp = CommonLogFormat
    Group = "coolsite"
    File = /data/logs/apache/coolsite/common_log
  }
  Log {
    RewriteTimestamp = CommonLogFormat
    Group = "slicksite"
    File = /data/logs/apache/slicksite/combined_log
  }
}
```

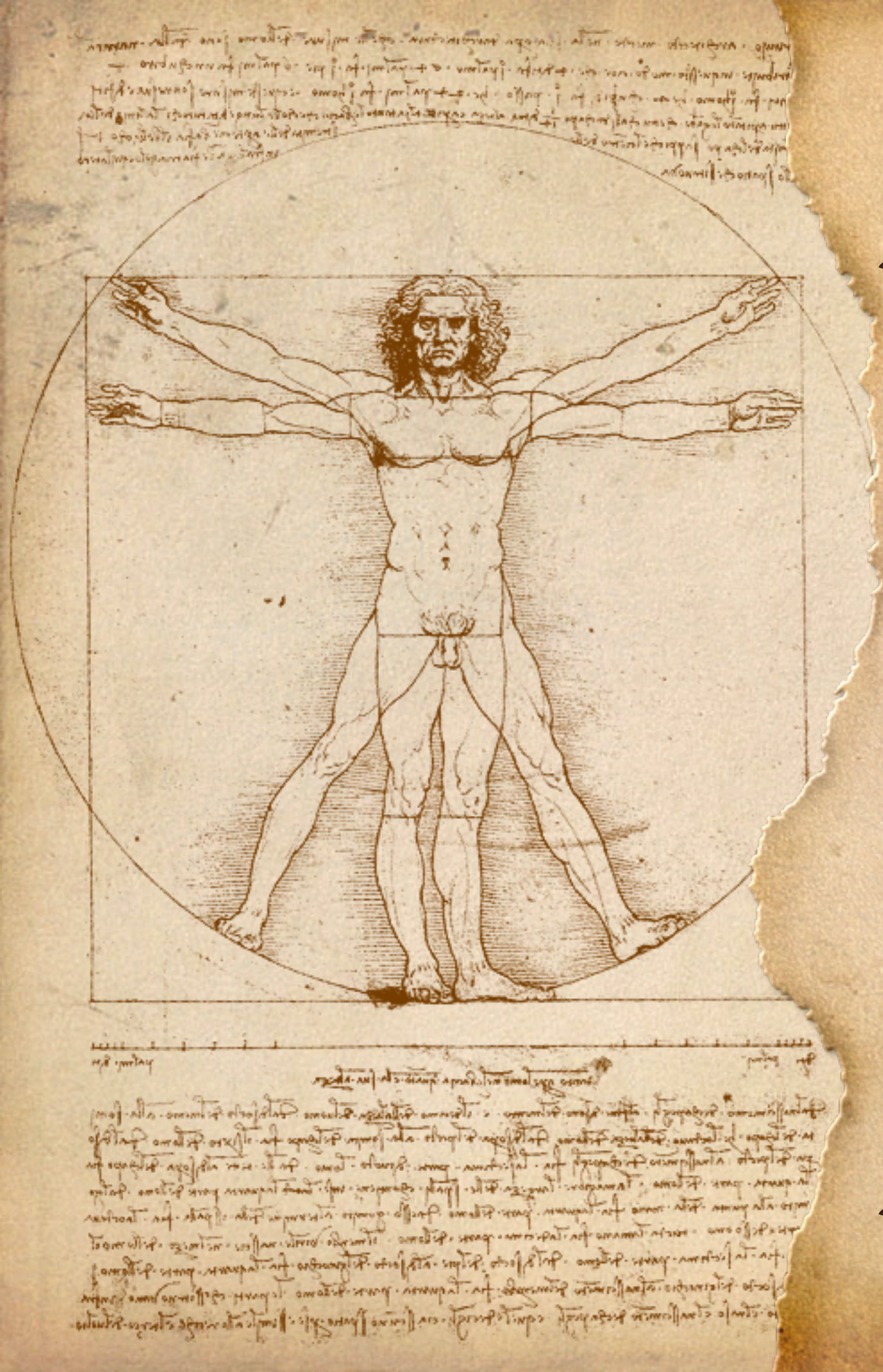
Spreadlogd: kung-fu (1)

```
BufferSize = 65536
PerlLib /opt/spreadlogd/custom
PerlUse mylogger
Spread {
  Port = 4913
  Log {
    RewriteTimestamp = CommonLogFormat
    Group = "coolsite"
    PerlLog mylogger::log
    File = /data/logs/apache/coolsite/common_log
  }
  Log {
    RewriteTimestamp = CommonLogFormat
    Group = "slicksite"
    File = /data/logs/apache/slicksite/combined_log
  }
}
```

Spreadlogd: kung-fu (2)

```
package mylogger;

use DBI;
our $dbh;
our $sth;

sub log($$$) {
    my $sender = shift;
    my $group = shift;
    my $message = shift;
    my ($user, $host) = ($sender =~ /#([^#]+)#([^#]+)/);
    chomp($message);

    $dbh ||= DBI->connect("DBI:mysql:database=weblogs", "logger", "",
                                                { RaiseError => 0 });
    warn "DBI->connect failed." unless($dbh);
    if($dbh) {
        $sth ||= $dbh->prepare(q{INSERT INTO logs (host, group, timestamp, data)
                                                VALUES(:1,:2,NOW(),:3)});
        $sth->execute($host, $group, $message);
    }
}
```

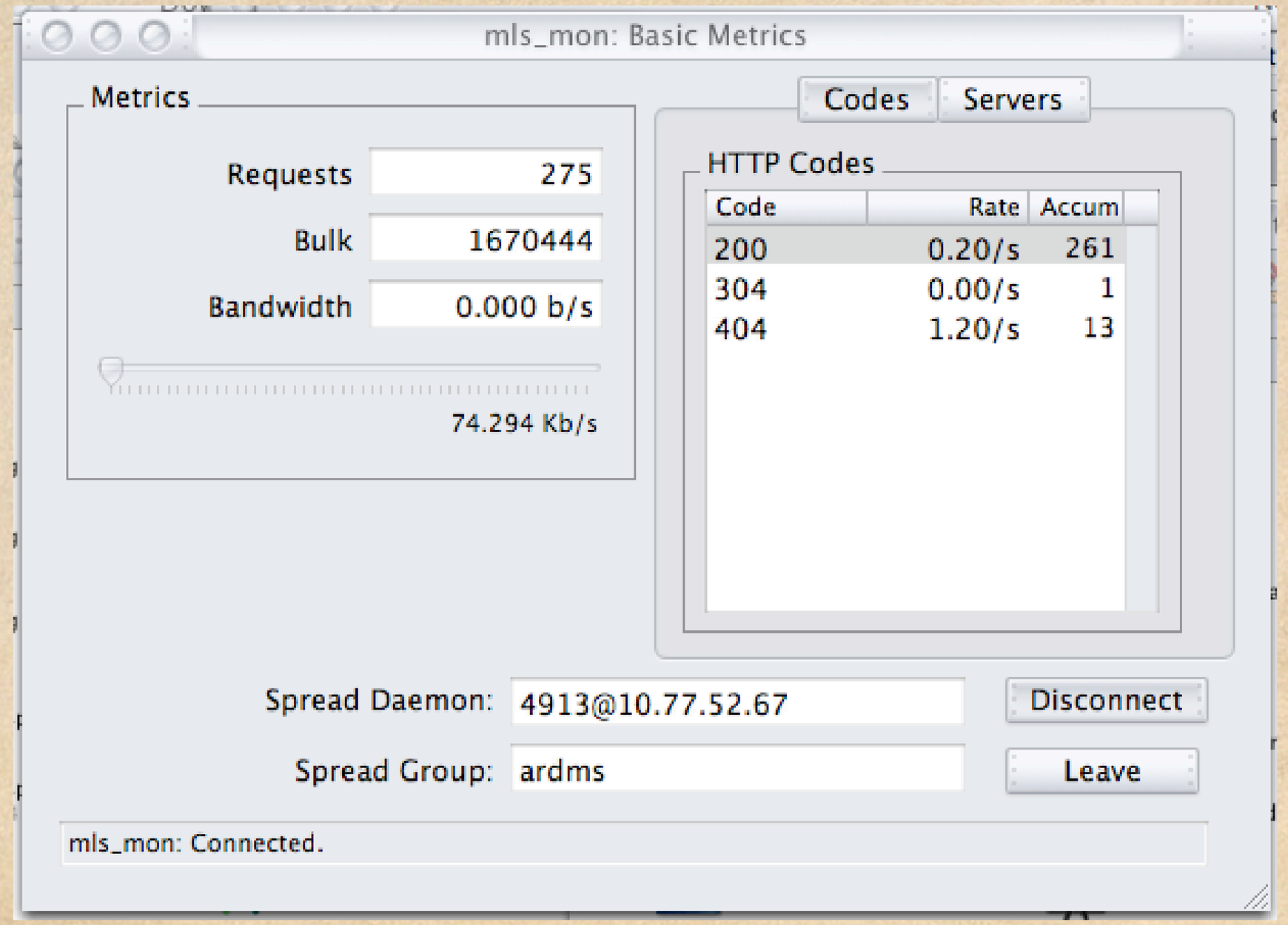
Understanding New Possibilities



Advances

- Logs are now streaming in real time
  - Real-time metrics
    - per server hit rates (traffic)
    - per server hits by response code
      - relative error serving rate
    - per server document size metrics
      - detect unexpected bugs do to anomalous traffic
  - Track deeper data
    - user habits
    - length of visit online
- All this happens passively

Stupid Pet Tricks



Credit Where Credit's Due

The John Hopkins University

The Center for Networking and Distributed Systems

OmniTI Computer Consulting

The Authors and Contributors of Spread:

Yair Amir, Michal Miskin-Amir, Jonathan Stanton, Christin Nita-Rotaru,
Theo Schlossnagle, Dan Schoenblum, John Schultz, Ryan Caudy, Ben Laurie,
Daniel Rall, Marc Zyngier

The Authors of mod_log_spread and Tools:

George Schlossnagle, Theo Schlossnagle, Jonathan Stanton, Yair Amir

Questions?