Forschungszentrum L3S

Webs of Trust in Distributed Environments

Bringing Trust to Email Communication

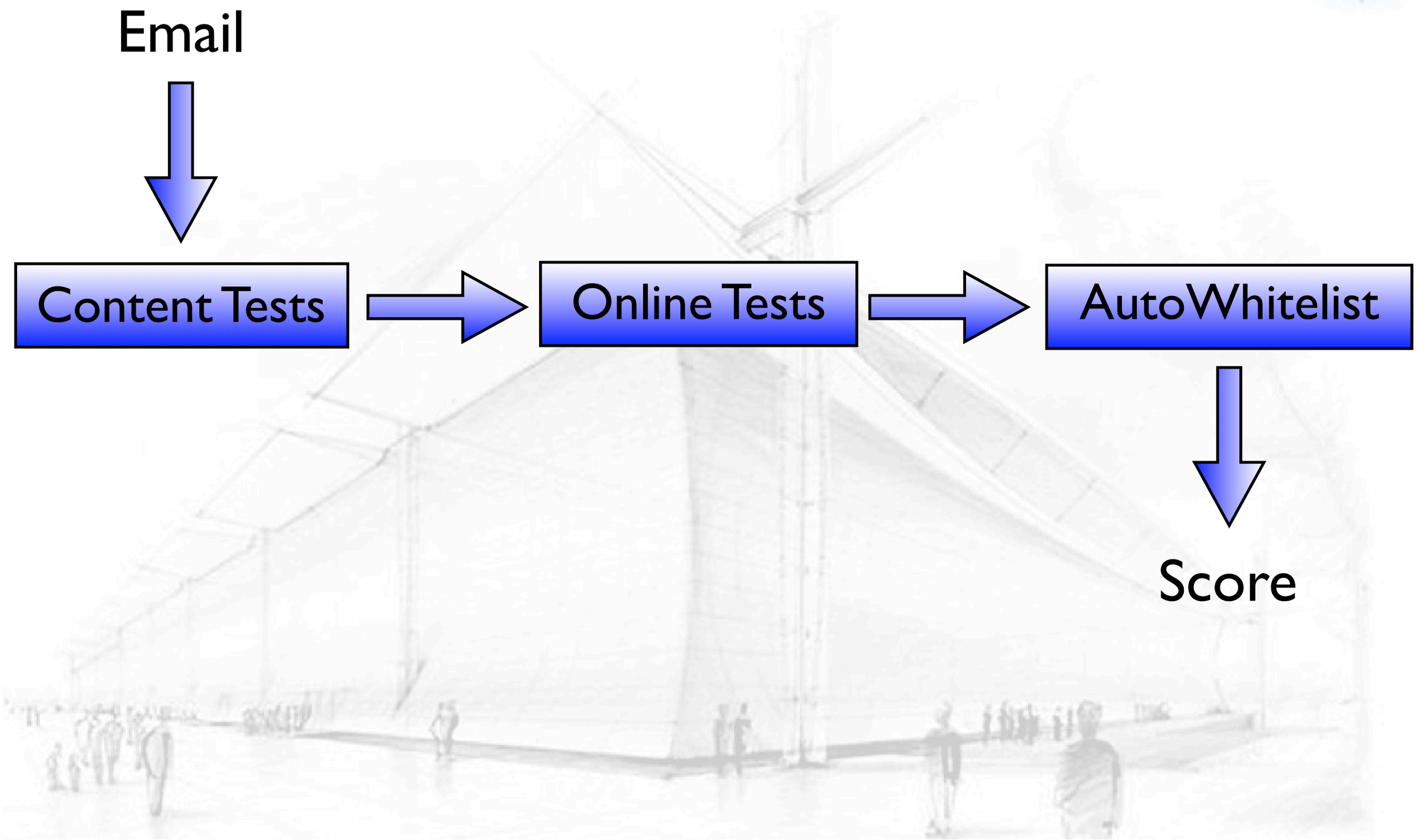
BSc. Presentation - Info-Lunch, 03.11.2004

Fighting Spam

Hmmm, tasty!!

Spamassassin

- Program for filtering unwanted Email messages
- Classifies Emails with scores as Spam or non-Spam
- Written in Perl and extensible


Email
Content Tests
Online Tests
AutoWhitelist
Score

Tests

- Header and text analysis: scanning for invalid headers, bad words (”Porn”) etc.
- Bayesian filtering: words or short sentences that often appear - filter “learns”
- DNS Blocklists: connections from a listed server are rejected
- Collaborative filtering databases: DCC, Razor

AutoWhitelist (AWL)

- Computes a score based on the history of a sender
- Consists of:
  1. The sender of an Email
  2. The IP of the Email server
  3. Number of Emails received from sender
  4. Total score for that sender

Scores in the AWL

$$MEAN = \frac{TOTAL}{COUNT}$$

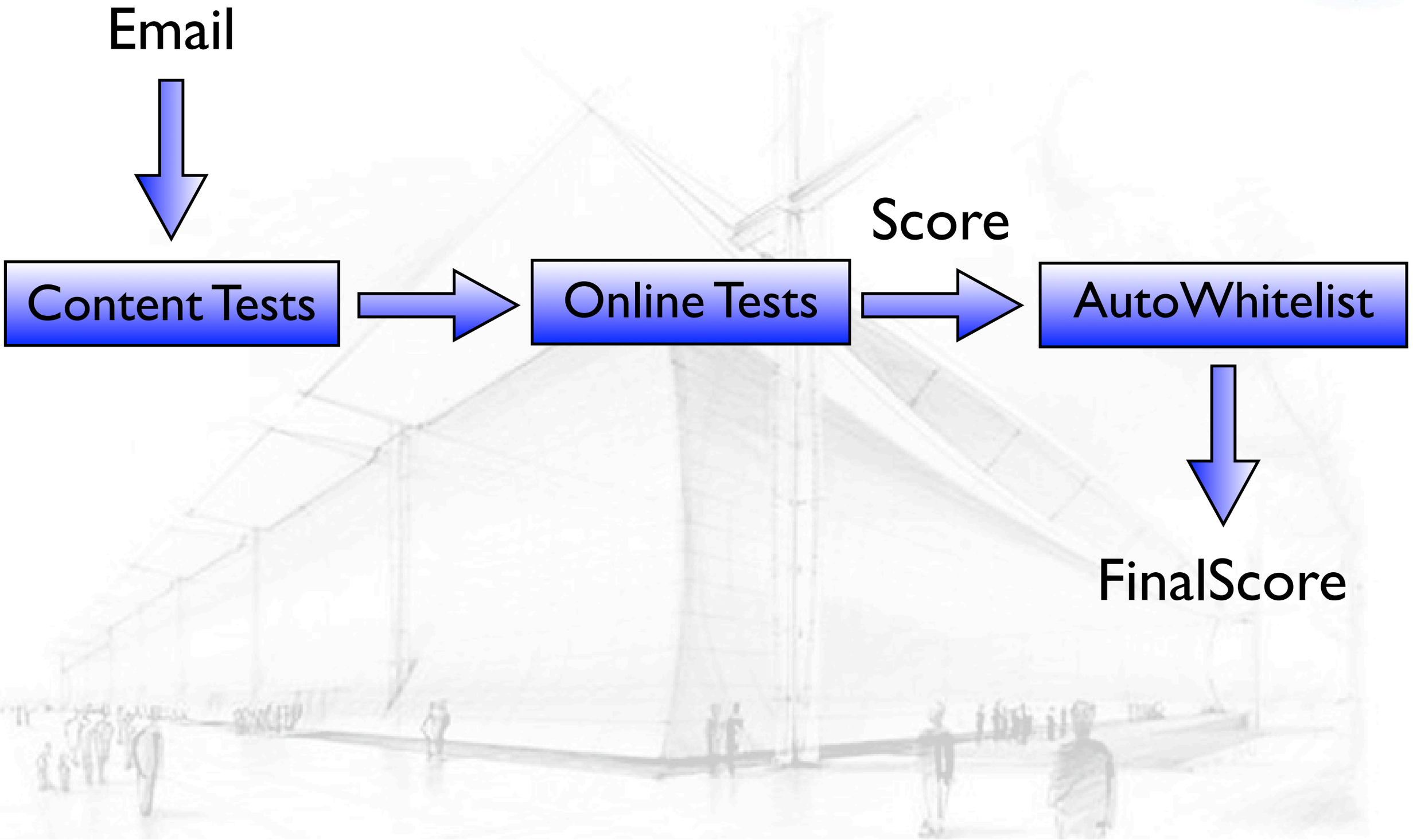
$$FINALSCORE = SCORE + (MEAN - SCORE) * FACTOR$$

Example:

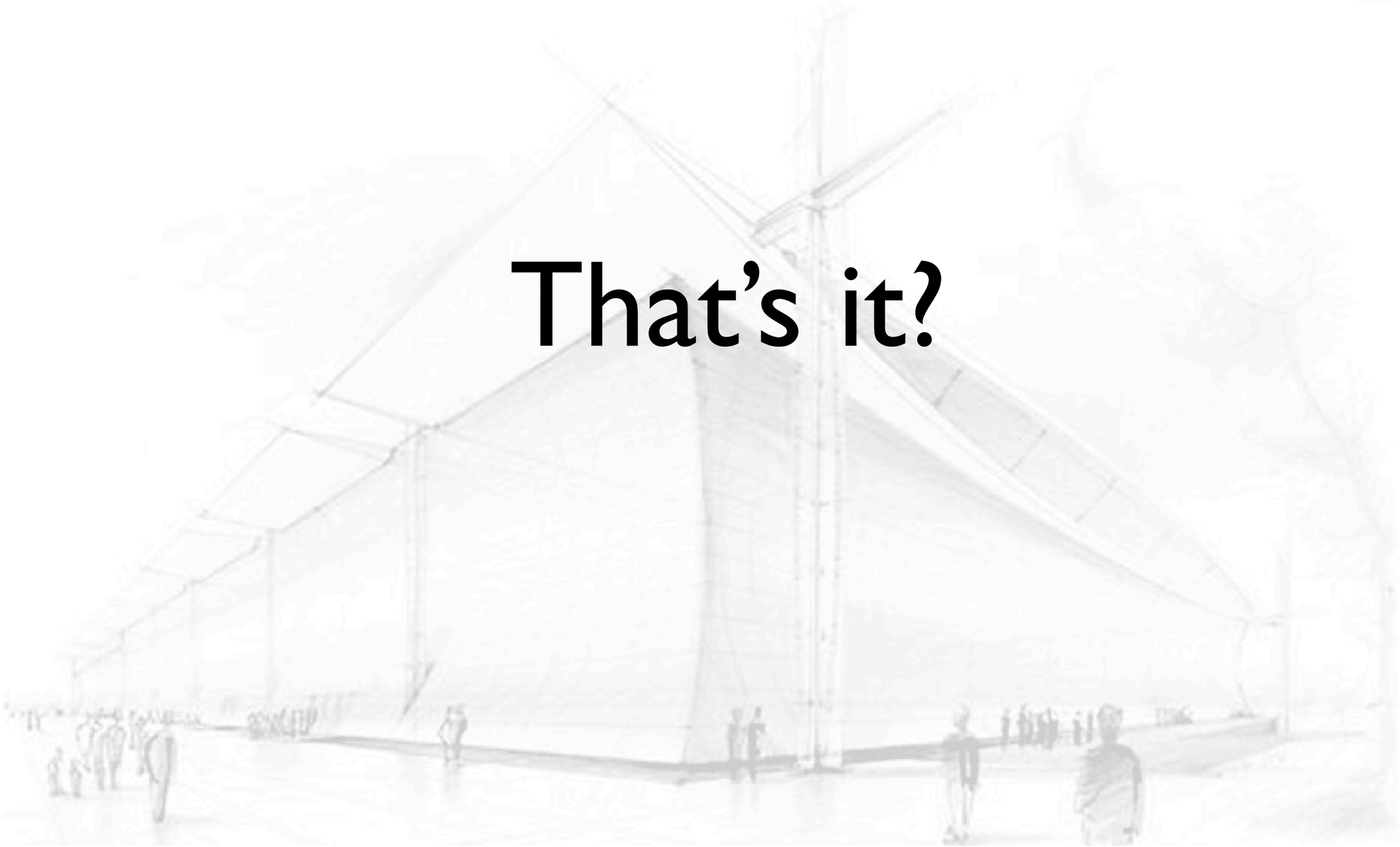
```
controller@club4x4.net|ip=82.49 2 37.628
```

New Email scores 20 Mean=18.814 Factor=0.5 (default)

Finalscore = 20 + (18.814-20) * 0.5 = 19.407


Email
Content Tests
Online Tests
Score
AutoWhitelist
FinalScore

That's it?

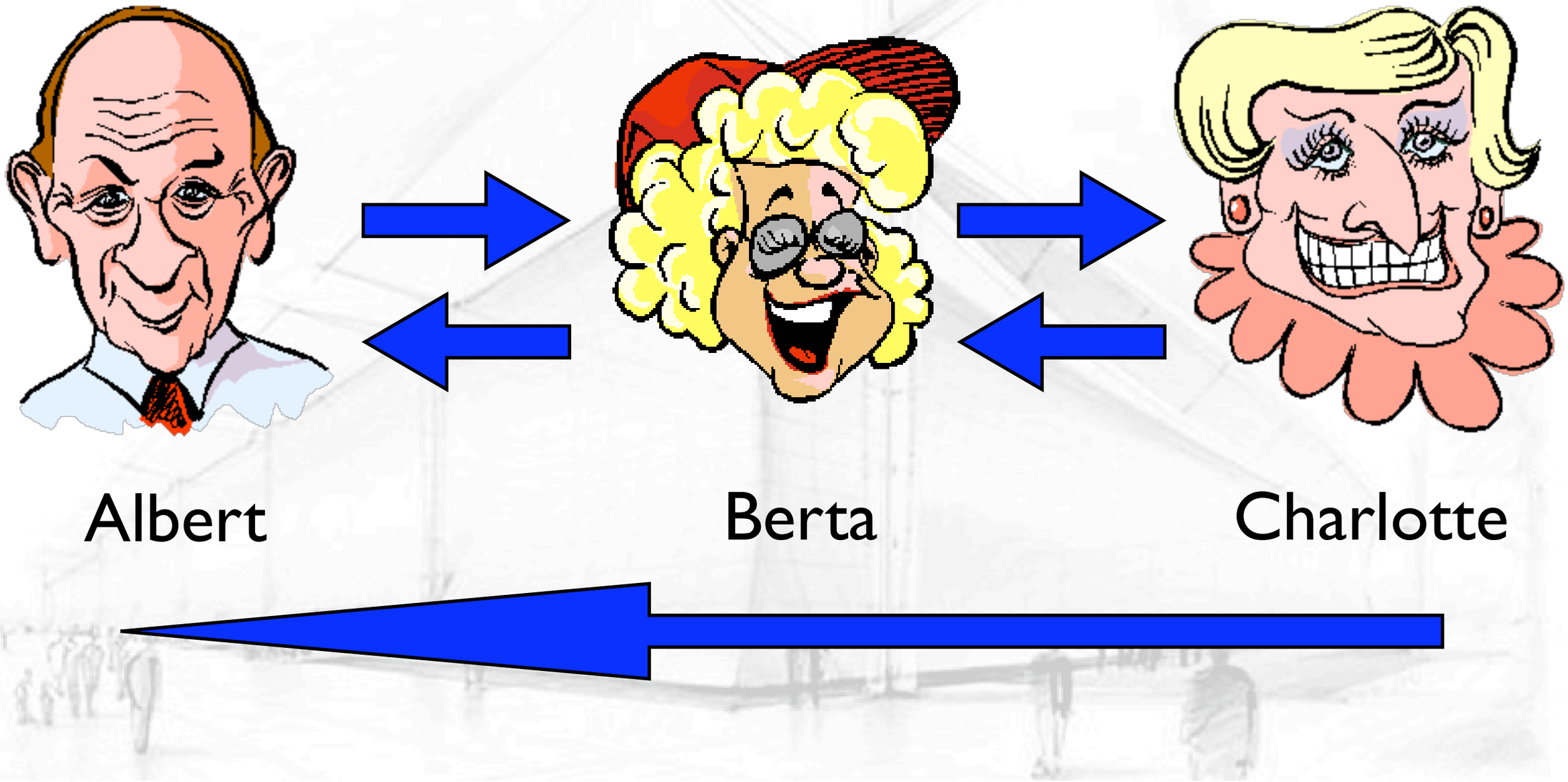


Need for Mailrank

False positives in SpamAssassin: an Email is tagged as spam, but it's actually not

Example: Emails from friend's friends

Emails from friend's friends



The Idea of Mailrank

Information from AWL

`controller@club4x4.net|ip=82.49 2 37.628`

Send Email address, IP, Count, Score
to a central server

From PageRank...

- informal: "*a page has a high rank if the sum of the ranks of it's backlinks is high*"
- exact: $$R'(u) = c \sum_{v \in B_u} \frac{R'(v)}{N_v} + cE(u)$$

... to Mailrank

- Given a set of users N_U, that "points" to a spam address *Spam*
- The Mailrank is given as:

$$MR(Spam) = c \sum_{U} \frac{MR(U)}{N_U}$$

Preliminary Version

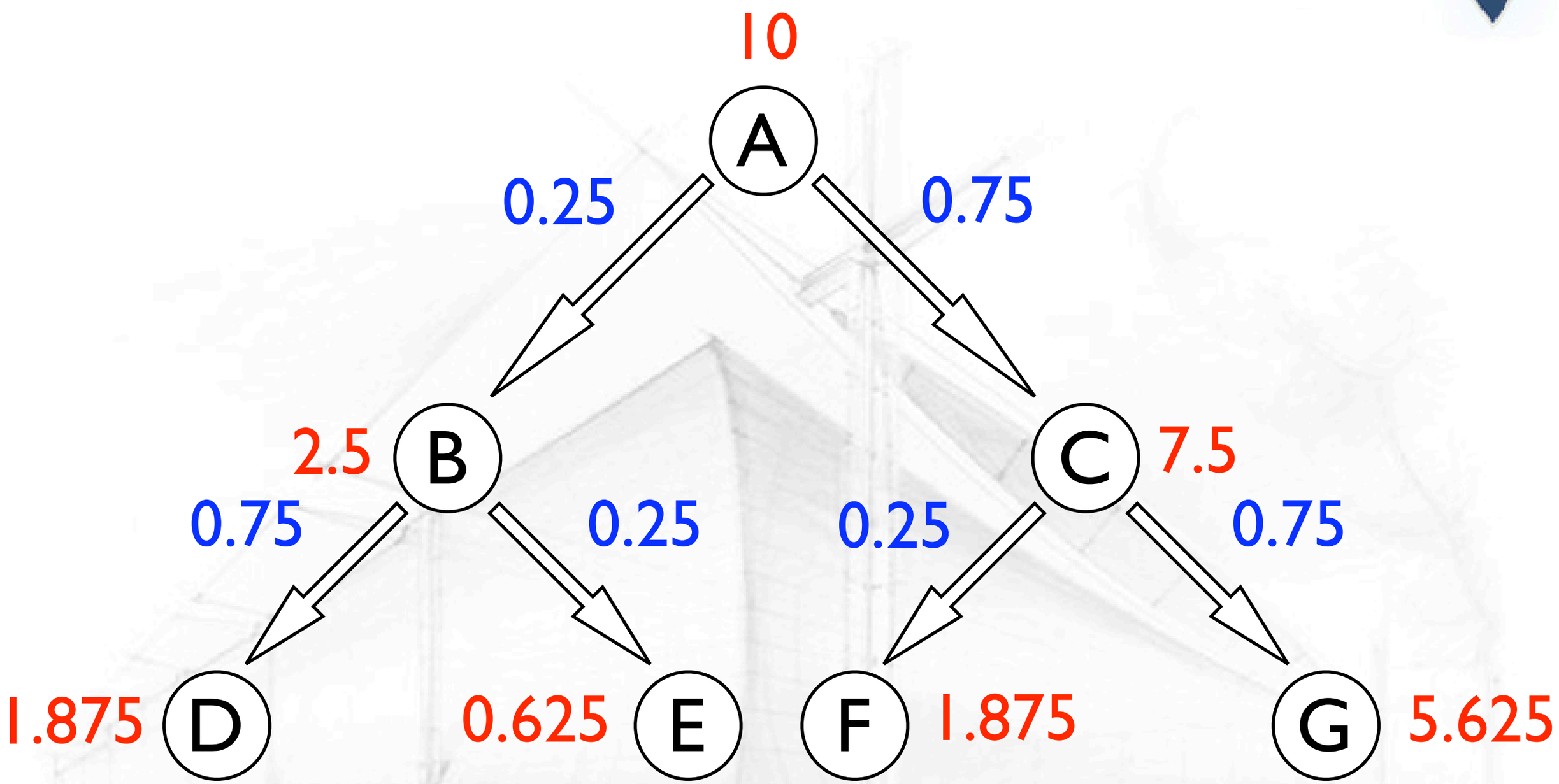
Using Mailrank

Examples

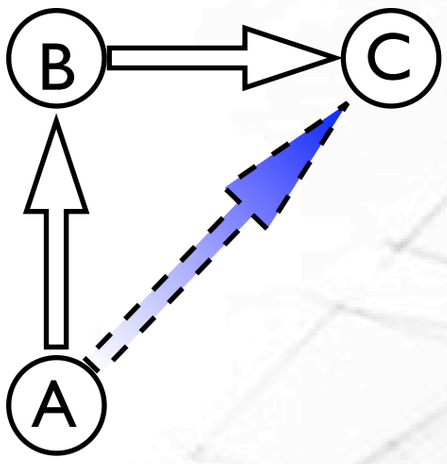
- If Mailrank is in the top 20% of all non-Spam Email addresses, add -5 to the Spam score
- If Mailrank is in the last 20% of all non-Spam Email addresses, add +10 to the Spam score

Ziegler/Lausen AppleSeed: Spreading Activation

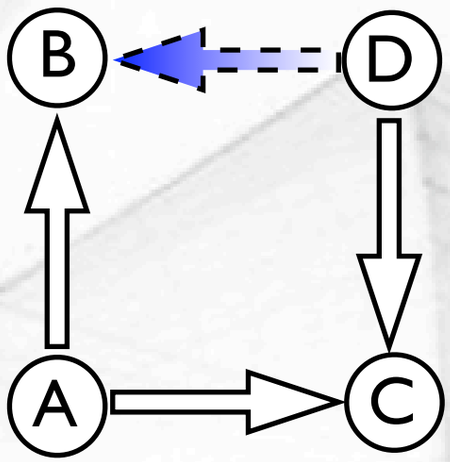
- Propagation of energy in a network
- Nodes are connected by edges
- Directed graph
- "Trust Decay": keep some trust in nodes
- Trust sinks: Backward propagation
- This is PageRank? No, Edges are weighted


10
A
0.25
0.75
2.5
B
C
7.5
0.75
0.25
0.25
0.75
1.875
D
0.625
E
F
1.875
G
5.625
Weights
Trustvalues

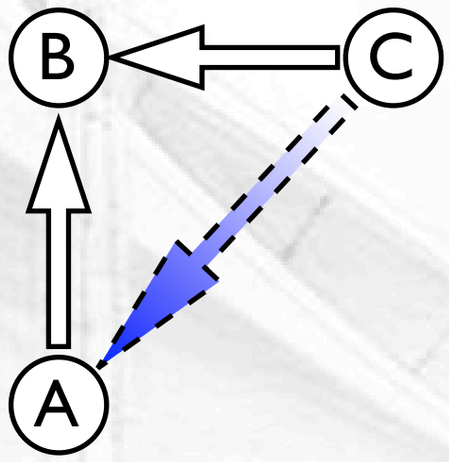
Guha: Trust/Distrust



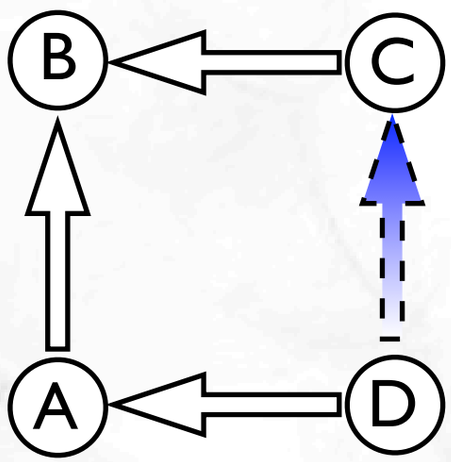
Direct Propagation



Co-Citation



Transpose Trust



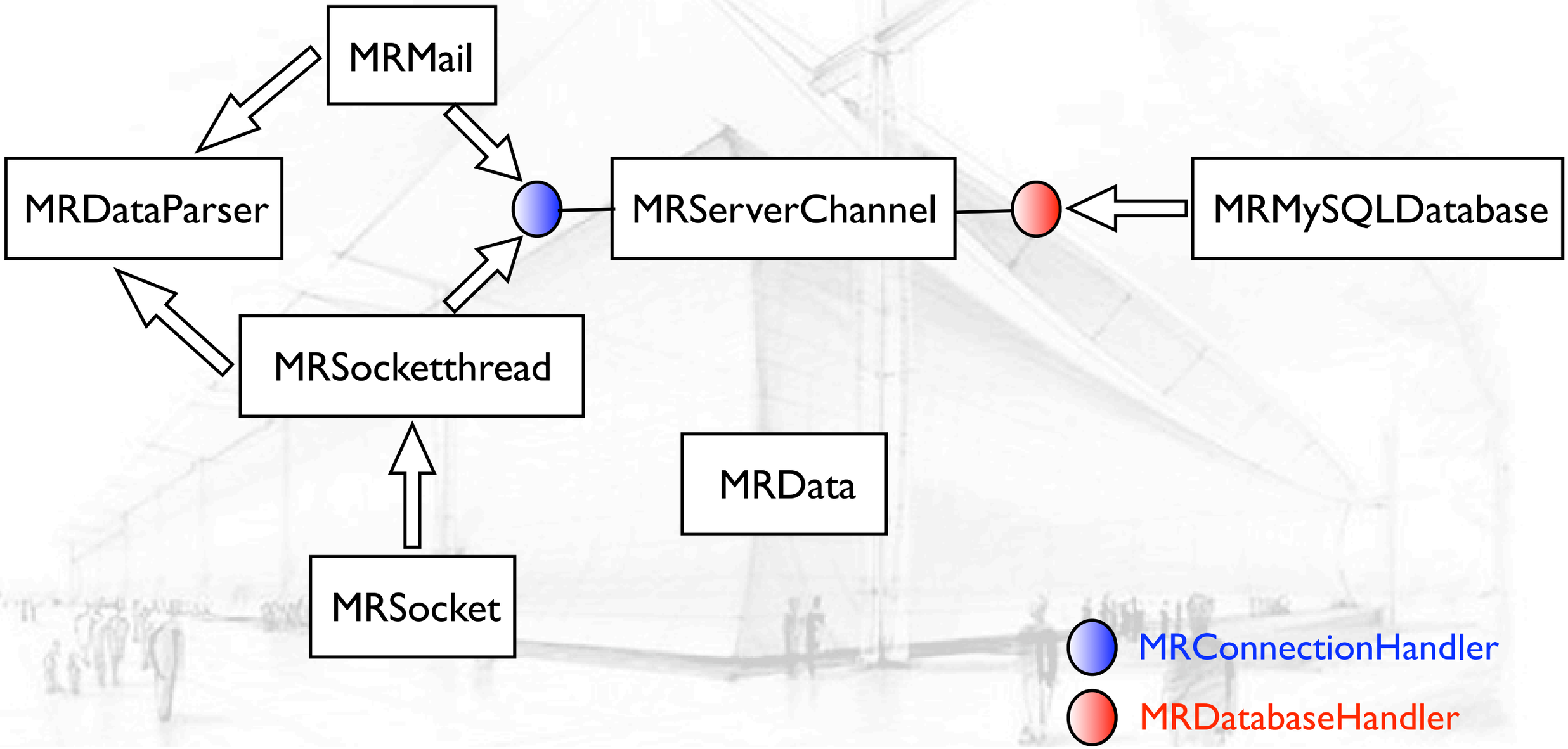
Trust Coupling

The Implementation

Design Goals

- Flexibility
- Abstraction
- Simplicity

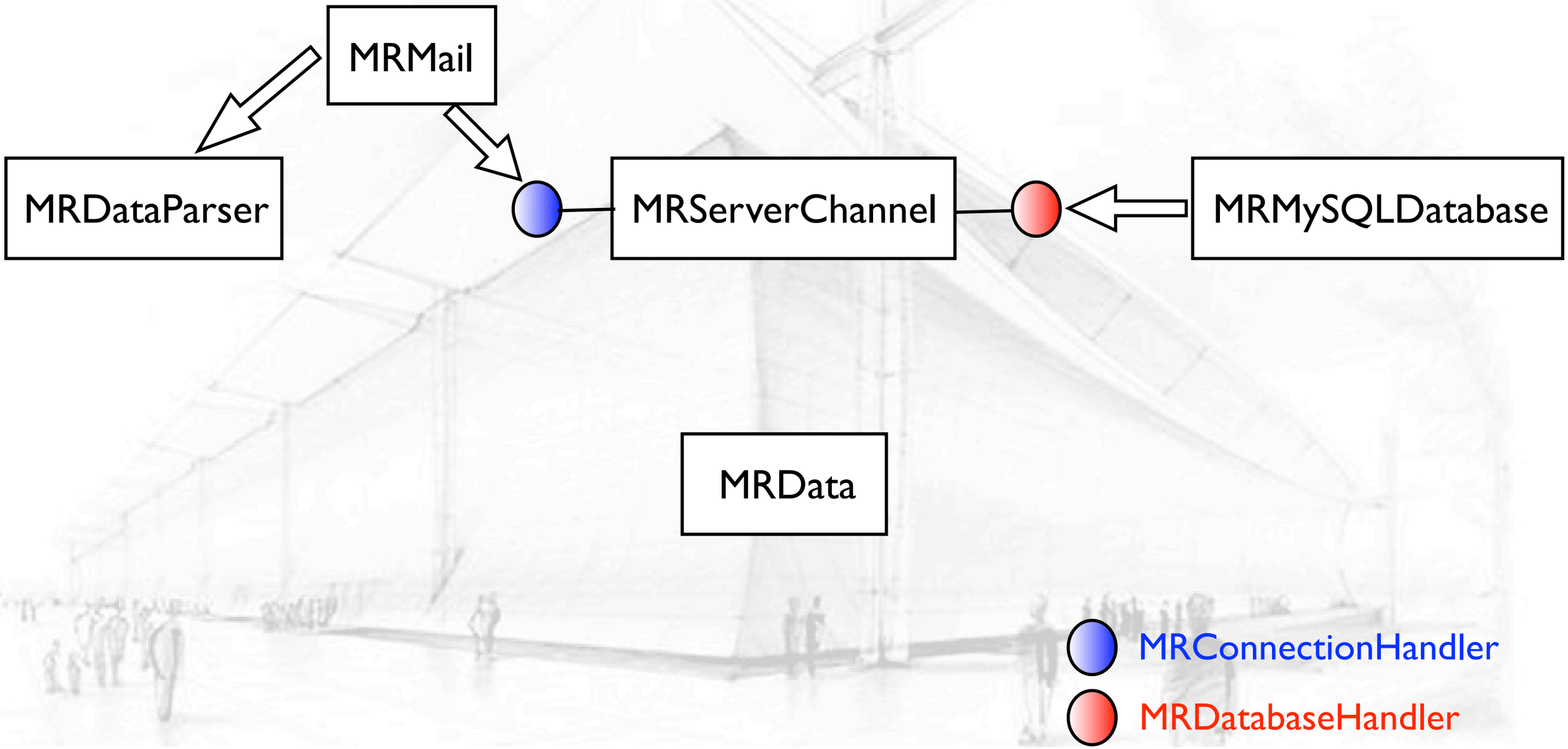
Overview


MRMail
MRDataParser
MRServerChannel
MRMySQLDatabase
MRSocketthread
MRData
MRSocket
MRConnectionHandler
MRDatabaseHandler

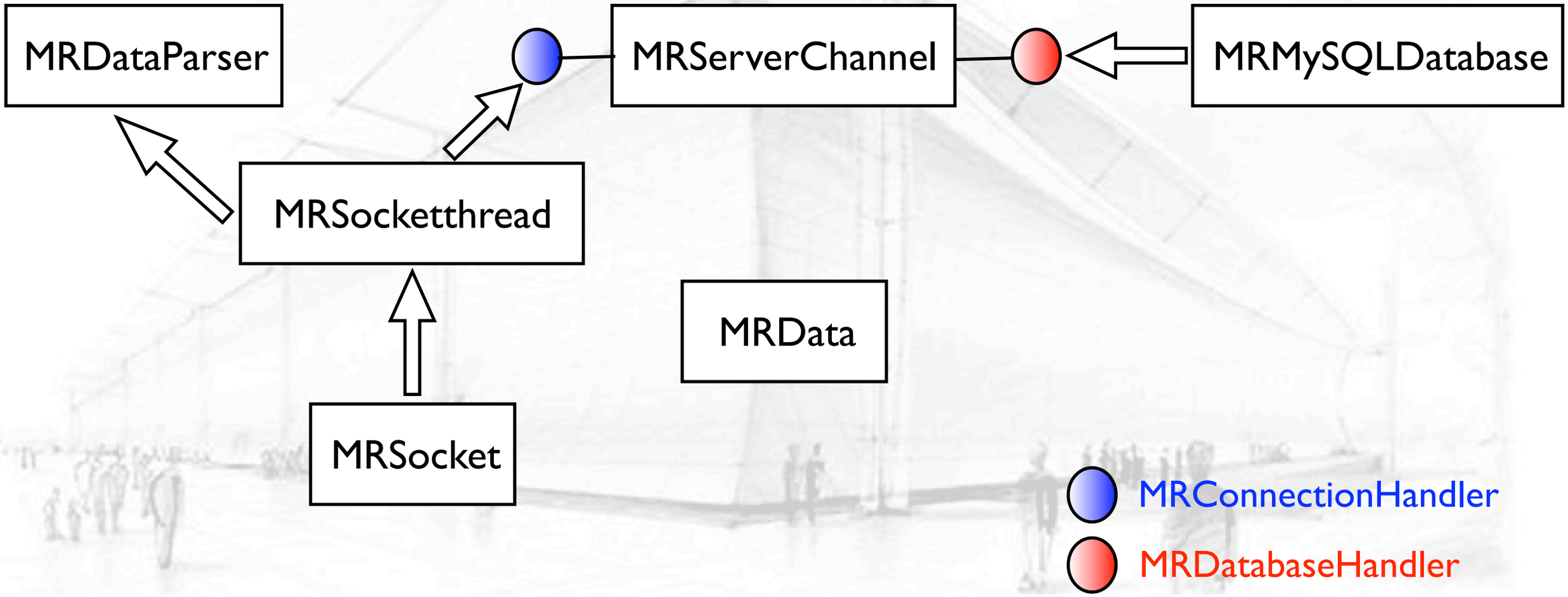
Abstraction: MRData

Fields in MRData
Command
Email address of user
Email address of AWL Entry
Score of AWL Entry
Count of AWL Entry

Mail


MRMail
MRDataParser
MRServerChannel
MRMySQLDatabase
MRData
MRConnectionHandler
MRDatabaseHandler

Socket


MRDataParser
MRServerChannel
MRMySQLDatabase
MRSocketthread
MRData
MRSocket
MRConnectionHandler
MRDatabaseHandler

Demo

What's next?

Further Work

- Develop the algorithm in detail
- Get the implementation done
- Provide a plug-in for SpamAssassin
- Paper (?)

Thanks! Questions?