Distributed Mail Servers

- **POST**: A secure, resilient, cooperative messaging system

- **Manageability, Availability and Performance in Procupine**: A Highly Scalable, Cluster-based Mail Service
  (Y. Saito, B. N. Bershad and H. M. Levy)
POST, ePOST

- ePOST
  SMTP-, IMAP-Server
- POST
  message storage, metadata, event notification
- PAST
  storage system
- Scribe
  multicast system
- Pastry
  P2P-Network
POST

3 Basic Services

- Secure Message Storage
- Event notification
- Metadata

- Robustness and Security
ePOST

- Email storage
- Email delivery
- Discussion
Procupine

- Goals
- Functional homogeneity
- hard / soft state
Procupine: Data Structures

- Mailbox Fragment (*hard state*)
- Mail Map (*soft state*)
- User Profile Database (*hard state*)
- User Profile Soft State (*soft state*)
- User Map (*soft state*)
- Cluster Membership List (*soft state*)
Procupine: Self Management

- Membership service
- Coordinator of a established Cluster
- Ways to discover failures
Procupine: Mail Traffic

- Mail Delivery
- Mail Retrieval
Procupine: Replication

- Update anywhere
- Eventual consistency
- Total update
- Lock free
- Ordering by loosely synchronized clocks
- Replicating mail fragments
Procupine: Load Balancing

- Distributed Load Balancing (functional homogeneity)
- Affinity-based scheduling (spread)
- Sending Replicated Mail
- Retrieving Replicated Mail
Distributed Mail Servers: Good Or Bad?

- **Positive**
  - No Bottleneck at a single Mailserver
  - No need for a big server
  - Scalability
  - More resilient
  - Availability

- **Negative**
  - Needs (at least one) reliable workstations (POST)
  - Heterogeneity cluster unrealistic in many cases
  - Security
  - Storage space in ePOST