Grapevine: An exercise in distributed computing

A. Birrell, R. Levin, R. Needham, M. Schroeder, Xerox PARC

CACM, April 1982

Grapevine

- Grapevine – Xerox PARC 1980
  - A loosely-coupled, distributed & replicated system
  - Provides *message delivery*, resource location, authentication, and access control
  - Operates in Xerox research internet – several Ethernet local networks, gateways and long distance links
- Design goals
  - No assumptions on message content
  - Cannot rely on the integrity of clients
  - Once the system accepts mail, it will be delivered
  - Fault tolerance to single computer failures
  - Decentralized administration

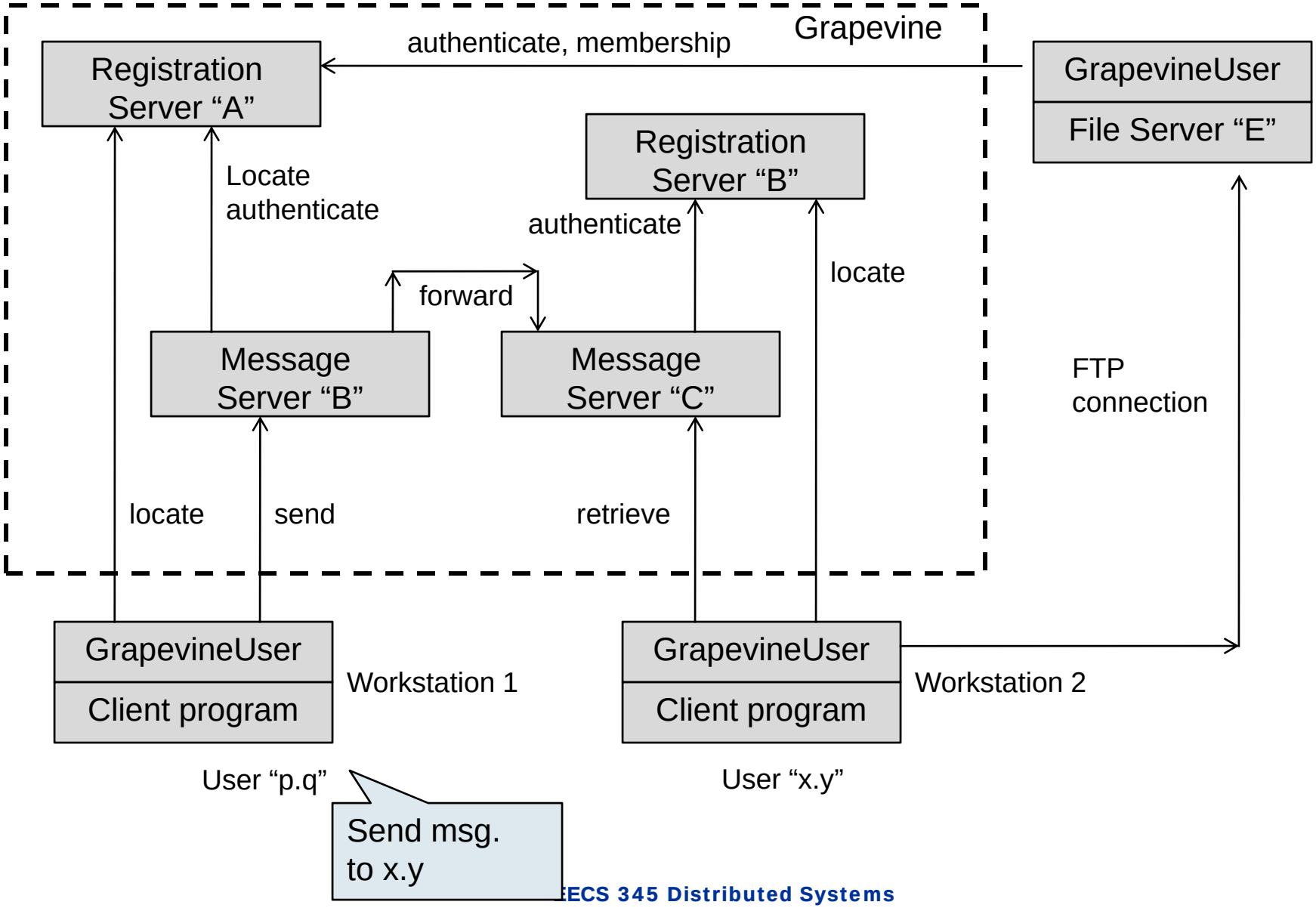
Grapevine ...

- Services provided by Grapevine divided into message and registration service
  - Each service a client of the other
  - Each server a combination of both message and registration server programs
- Message service
  - Accept msgs prepare by clients for delviery to individual or distribution lists
  - Msgs are buffered in inboxes on msg servers until the recipient request them
  - Each message server accepts retrieval requests for inboxes on that server (users have inboxes on at least 2 message servers)
  - Message need not be human readable text

Grapevine ...

- Registration service
  - Provides naming, authentication, access control & resource location to clients
  - Based on a registration database mapping names to information about users, machines, distribution lists, & acls
  - Entries can be for individuals or groups
    - Name of an entry – RName, plus password, ordered list of inbox sites, …
    - Group – a set of RNames for distribution lists or access control lists
  - Name space structured as a two-level hierarchy
  - Registration database is distributed and replicated at the level of an entire registry
  - Any registration server holding a registry replica can accept changes to it and becomes responsible for propagating it
- Grapevine client use the GrapevienUser package to get service

Grapevine: Functional diagram



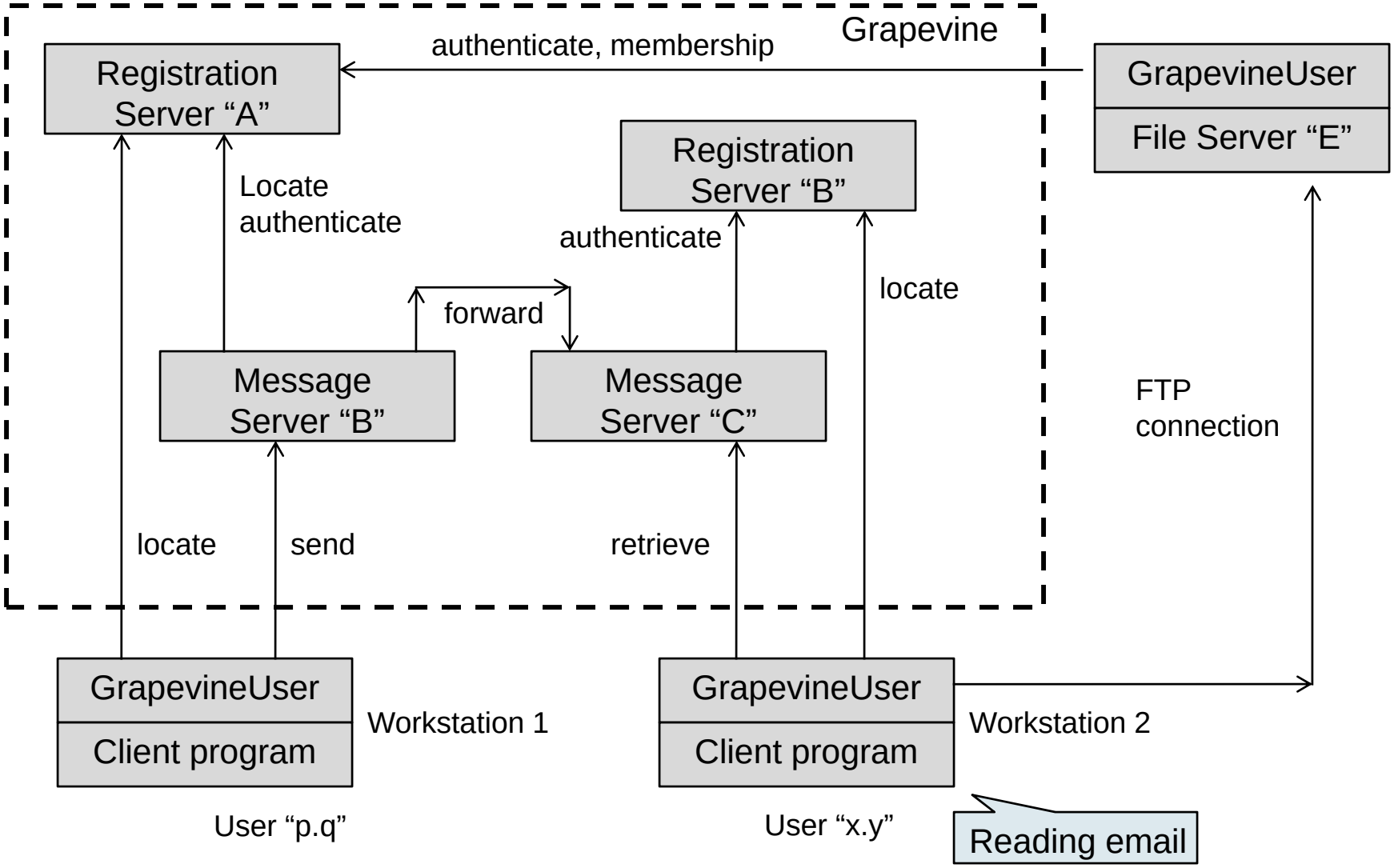
Grapevine: Sending a message

- User prepares message using mail client
- Mail client contacts GrapevineUser package on same workstation to send message
- GrapevineUser package
  - Contacts *any* Registration Server to get a list of Message Servers
  - Contacts *any* Message Server to transmit message
    - Presents source and destination userids, and source password, for authentication
      - Message Server uses *any* Registration Server to authenticate
    - Sends message body to Message Server
      - Message Server places it in stable storage & acknowledges receipt

Grapevine: Transport and buffering

- For each recipient, Message Server contacts *any* Registration Server to obtain list of Message Servers holding mail for that recipient
- Sends copy of the message to one of those Message Servers for that recipient (preferred site)
  - Hints for speeding up mappings
- If message cannot be deliver to some, reports this back to sender with an explanation

Grapevine: Functional diagram



Grapevine: Retrieving mail

- User uses mail client to contact GrapevineUser package on same workstation to retrieve mail
- GrapevineUser package
  - Contacts *any* Registration Server to get a list of each Message Server holding mail for the user ("inbox site")
  - Contacts each of these Message Servers to retrieve mail
    - Presents user credentials
      - Message Server uses any Registration Server to authenticate
    - Acknowledges receipt of messages so that the server can delete them from its storage

Grapevine: Scalability

- Registration database splint into registries
- Can add more Registration Servers
  - Eventual consistency in registration data base
  - Long term inconsistency solved with periodic anti-entropy mechanism
- Can add more Message Servers
- One thing didn't scale – handling of distribution lists
  - The accepting Message Server was responsible for expanding the list (recursively if necessary) and delivering to an appropriate Message Server for each recipient
  - Some distribution lists contained essentially the entire user community
  - *A better approach?*

Grapevine: Reliability

- Reliability through replication
- Don’t overlook the fact that using redundancy to achieve reliability requires the system to *always* have the spare resources
- Grapevine used archiving to deal with small disks in servers
  - Every night, messages +seven-day older were archived to a nearby file server
  - Problem – you couldn’t get your inbox if you cannot access the archive

Grapevine: Transparency

- Most of the time it looked as a centralized server with lots of storage and fast network links
- But every now and then, some surprises, e.g.
  - Delays in propagation of registration database changes
  - Harder for a sys admin to check if a distribution list is being used
  - Deleting names is always an issue – nothing like absolute, instantaneous delete
  - Deleting a message is not different
  - Expanding a distribution list of remote recipients can take a while
  - The effect of administration actions *must* be obvious to administrators
    - E.g. change the inbox site list used to required re-mailing an inbox

Summary

- A good experiment in building distributed systems
- It raised many of the issues we are still dealing with
  - Transparency – how much and at what cost?
  - Reliability as a key goal
  - How to best operate a dispersed system?
  - How to handle increasing, unexpected load?