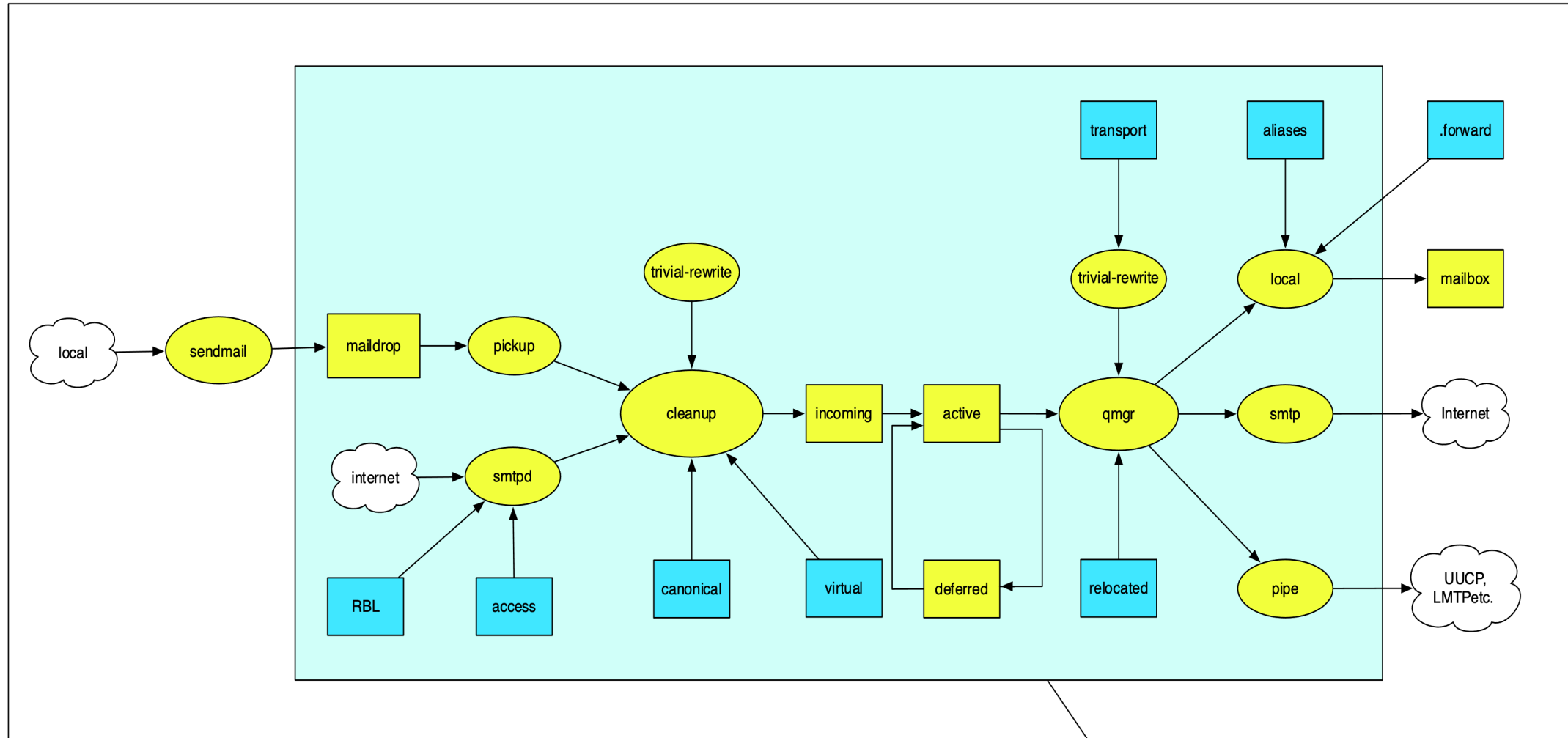# CS-297 PRESENTATION 3

Week of November 7

- Ajita Shrivastava (016523104)

Postfix – Architecture

Legend:

| Lookup table | Mail programs | Mail queues or files |

Controlled by Postfix master daemon

Postfix, like many software applications, is programmed following a modular and layered architecture that separates different components responsible for various aspects of email handling. It is written in C and follows a highly modular design to enhance maintainability and performance. Below is a simplified overview of the architecture of Postfix from a programming perspective:

1. *Main Program (postfix/main.c):* The main program is the starting point of the Postfix application. It sets up the environment, parses the configuration files, and initializes various subsystems. It then enters the main event loop.

2. *Configuration (postfix/src/global/mail_params.c):* Postfix reads its configuration from various files (e.g., main.cf), and this module parses and processes those configurations. Configuration values are used to customize the behavior of Postfix.

3. *SMTP Server (postfix/src/smtpd):* This is a critical component for receiving and handling incoming email. The SMTP server listens for incoming email connections on port 25 (or other configured ports) and processes incoming SMTP commands from other mail servers. It is composed of several modules, such as the smtpd, smtp, and smtp_stream modules.

4. *Message Queuing (postfix/src/qmgr):* The queue manager is responsible for managing the mail queue. It receives messages from the SMTP server and schedules them for delivery. It ensures that messages are delivered reliably, even in cases of temporary delivery issues.

5. ***Local Mail Delivery (postfix/src/local):*** The local delivery component is responsible for delivering email to local user mailboxes.

6. ***Message Transfer (postfix/src/transport):*** This module handles the routing and delivery of messages to their final destinations. It interacts with DNS to determine the target mail servers for external email delivery.

7. ***Logging (postfix/src/global/maillog.c):*** Postfix includes a robust logging system to record various events and activities. This is crucial for monitoring and debugging.

8. ***Authentication and Security (postfix/src/smtpd):*** Postfix includes authentication mechanisms to ensure that only authorized users and servers can send and receive email. It also supports encryption protocols like TLS to secure email transmission.

9. ***Database Lookups (postfix/src/global/dict_*.c):*** Postfix can be configured to use external databases for certain tasks, such as recipient validation, sender access control, or address rewriting. Various dict_* modules provide the interface to these external databases.

Dovecot – Architecture
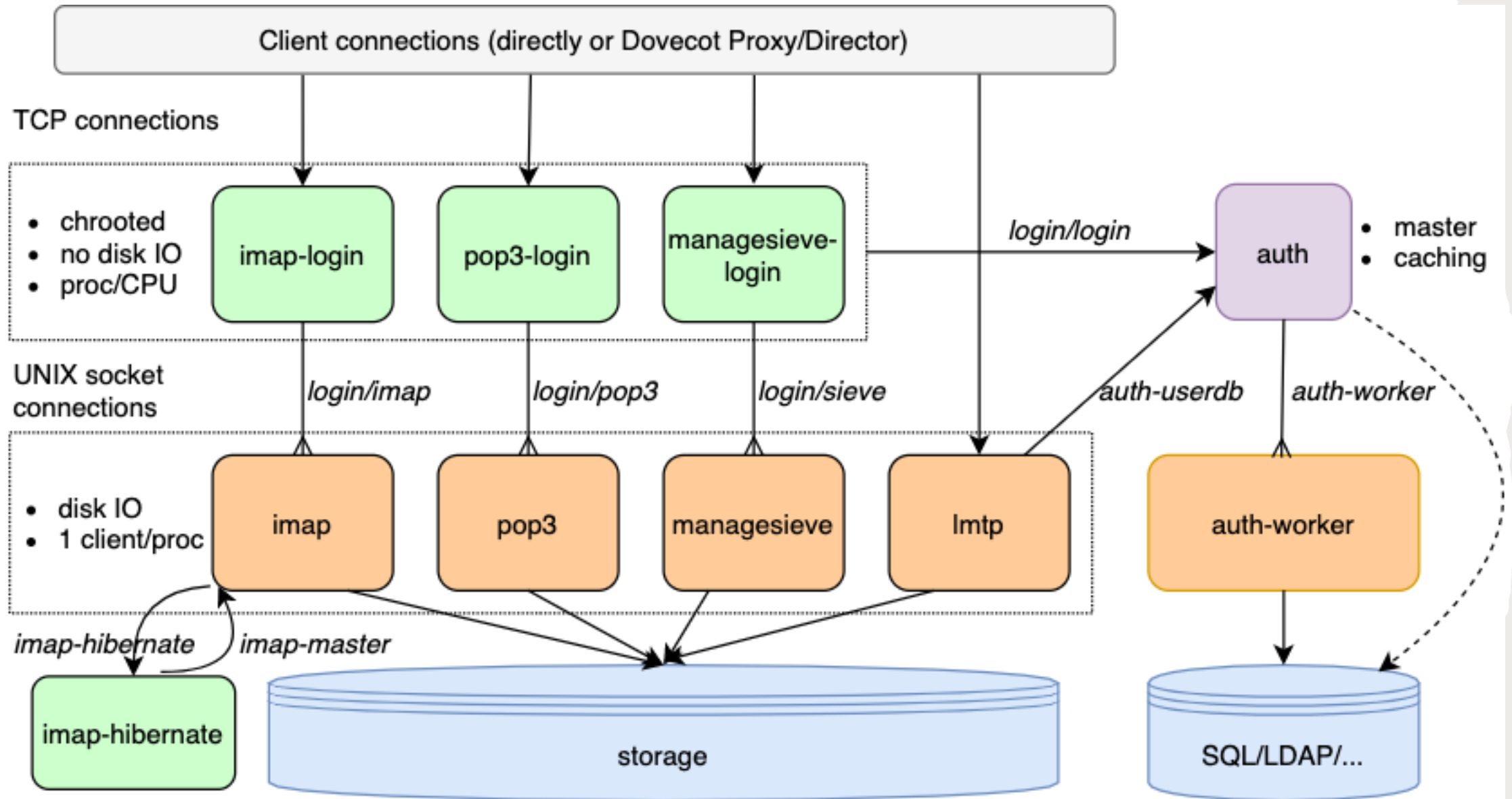
Dovecot processes

1. ***Master process (dovecot):*** This process keeps all the other processes running. If a child process dies, another one is restarted automatically if necessary. The master process runs as root, so its functionality is attempted to be kept minimal.

2. ***Log process (log):*** The Dovecot master process sets up a separate pipe for each service, which is shared by all the processes of that service. The write side of the pipe becomes the processes' stderr fd, while the read side is read by the log process.

3. ***Config process (config):*** The config processes parse the configuration file and feed the parsed output in a simplified format to all the other processes via UNIX socket connections. The master process reads its configuration in a different way: It first executes the doveconf binary, which reads the configuration into environment variables and then executes back the dovecot master binary.

4. ***Authentication process (auth):*** The auth process handles everything related to the actual authentication: SASL authentication mechanisms, looking up and verifying the passwords and looking up user information.

5. ***Login processes (imap–login, pop3–login):*** The login processes implement the required minimum of the IMAP, POP3, ManageSieve or Submission protocols before a user logs in successfully. Each protocol is handled by a separate process (and binary).

6. ***Mail processes (imap, pop3, lmtp):*** These processes handle the actual post–login mail handling using the privileges of the logged in user.