­­МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

­­­

Розрахункова робота №1

«Клієнтські системи електронної пошти на базі продуктів з відкритим кодом»

з курсу «’Комп’ютерні мережі»

для студентів базового напрямку 6.08.04 "Комп’ютерні науки"

(заочна форма навчання)

Варіант 10

Виконав студент гр. КНз-3

Чалий Михайло

­­

Львів 2014

## Мета роботи

**Метою** розрахункової роботи є дослідження заданої технології та проектування вирішення в галузі КМ на базі цієї технології.

## Теоретичні відомості

[Електронна пошта](http://uk.wikipedia.org/wiki/%D0%95%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%B0_%D0%BF%D0%BE%D1%88%D1%82%D0%B0) - є одним із основних і найпопулярніших сервісів у комп'ютерних мережах. Програму електронної пошти первісно було створено для передавання текстових повідомлень, але завдяки зручності цього сервісу виникла потреба в розширенні його функціональних можливостей. З повідомленнями електронної пошти можна передавати не лише текст, але й вкладення, якими можуть бути майже будь-які файли (фактично обмеження накладаються лише на розмір вкладень).  
У корпоративних мережах система електронної пошти забезпечує переважно внутрішній та зовнішній обмін інформацією. Але у багатьох випадках функції цього сервісу є набагато більшими. Електронну пошту можна використовувати, як компонент системи документообігу і як транспортний протокол корпоративних застосувань або як засіб утворення інфраструктури електронної комерції.  
Щоб мати змогу користуватися електронною поштою, необхідно створити на певному сервері поштову скриньку, яка здебільшого реалізується у вигляді одного великого текстового файлу (поштового файлу). Доставлення повідомлення користувачу полягає у дописуванні цього повідомлення в кінець файлу. для того щоб прочитати повідомлення, користувачу необхідно застосувати спеціальне програмне забезпечення.  
Головною проблемою в реалізації системи електронної пошти є доставлення повідомлень, яке передбачає модифікацію файлу користувача - одержувача повідомлення. Відправлення повідомлень і їх передавання мережею може бути здійснене за різними протоколами, але найчастіше використовують [SMTP](http://ru.wikipedia.org/wiki/SMTP) (Smple Mail Transfer Protocol - простий протокол передавання пошти). Цей протокол реалізують поштові SMTP-сервери. Як альтернативу можна назвати протокол UUCP.  
Функціонально програма Sendmail складається з трьох відокремлених компонентів:

* агент користувача - дає змогу користувачу формувати повідомлення для відправлення і декодувати отримані повідомлення, які знаходяться у поштовій скоринці;
* агент пересилання - відповідає за приймання і пересилання кореспонденції з одного поштового сервера на інший;
* агент доставляння - керує поштовою скринькою користувача, додаючи до неї повідомлення що надходять.

Найголовнішим компонентам системи електронної пошти є агент пересилання. Функції інших компонентів можуть виконувати сторонні програми. В ролі агента пересилання замість Sendmail можна також використовувати інші програми всі вони мають спільну назву агенти пересилання пошти (Mail Transer Agents, MTA).  
В електронній пошті є і інший підхід. Користувач застосовує спеціальну програму - поштового клієнта, що здійснює доступ до сервера за спеціальними протоколами читання електронної пошти, серед яких найпопулярніші [POP3](http://uk.wikipedia.org/wiki/POP3) ( Post Office Protocol) та [IMAP](http://uk.wikipedia.org/wiki/IMAP) (Internet Mail Access Protocol). У цьому випадку може бути реалізований один з таких сценаріїв:

* користувач забирає з сервера всю пошту, а потім сортує, обробляє та зберігає її на своєму клієнтському комп’ютері;
* користувач сортує, обробляє, зберігає свою пошту на сервері, а його програма-клієнт лише здійснює керування.

Протокол POP3 призначений для реалізації першого сценарію, IMAP для другого.  
В пересиланні та доставлянні електронної пошти ключову роль відіграють програми-агенти, спілкування з якими (і їх спілкування між собою) здійснюється за протоколом SMTP. Даний протокол передбачає дуже просту ідентифікацію і не передбачає автентифікації. Тобто, агенти пересилання пошти фактично не перевіряють, хто цю пошту передає. На час створення протоколу це видавалося прийнятним, оскільки, протокол SMTP передбачає саме передавання пошти і тому спроби неавторизованого доступу за цим протоколом ніяк не порушують конфіденційності та цілісності даних. Але з розвитком [Інтернету](http://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82) почали з'являтися проблеми:

* засмічення електронної пошти некорисними і небажаними повідомленнями;
* розсилання повідомлень, що містять шкідливі програми;
* відправлення повідомлень від інших користувачів.

## SMTP

**SMTP** - це поштовий протокол хост-хост. Стосовно протоколу SMTP не вживають терміни "клієнт" і "сервер". У цьому випадку використовують поняття "відправник" і "одержувач". SMTP-сервери можуть виступати в ролі як клієнта, так і сервера.  
За стандартом SMTP-сервер використовує 25-й порт [TCP](http://uk.wikipedia.org/wiki/TCP). Підключитися до нього можна за допомогою клієнта [Telnet](http://uk.wikipedia.org/wiki/Telnet). Після встановлення SMTP-з'єднання можна починати передавати поштове повідомлення. Передавання кожного повідомлення здійснюється SMTP-транзакцією. Щоб передати повідомлення, спочатку необхідно створити зворотну адресу (тобто, від кого це повідомлення) за допомогою команди "MAIL FROM", яка відкриває транзакцію.   
Далі йде команда "RCPT TO" в якій вказують адресу одержувача. Це другий крок транзакції: визначення адреси призначення повідомлення. Якщо один лист адресований кільком одержувачам, команду "RCPT TO" потрібно повторити необхідну кількість разів.  
Після цього видається команда "DATA", яка дає вказівку серверу перейти серверу у режим приймання тексту повідомлення.  
Передбачається, що клієнт правильно назвав себе у команді "HELO" (команда автентифікації). Але під час спілкування з багатьма серверами у привітанні після "HELO" можна написати будь-що: сервер може взагалі ігнорувати введену адресу вузла( домену чи ІР) і використовувати автоматично визначне доменне ім'я комп'ютера користувача. Поведінка різних серверів залежить від їх настроювань, які реалізують обрану політику безпеки.

## РОР3

**РОР3** - найпопулярніший протокол читання електронної пошти з сервера. Він підтримує деякі прості команди, які дають змогу клієнту базові операції керування поштовою скринькою і завантажувати з поштового сервера листи, отримані від іншого поштового сервера. Обробленням повідомлень протокол не займається - це робить клієнтське програмне забезпечення. Також даний протокол не відповідає за відправлення пошти - цим займаються інші протоколи (здебільшого SMTP).  
Сервер РОР3 за стандартом використовує 110 порт, хоча це можуть бути і інші порти. Поштові клієнти дають змогу задати номер порту для кожного сервера. Для взаємодії з РОР-сервером інколи використовують протокол Telnet. Сервер коректно відпрацьовує такі команди, звісно, якщо вони правильні..  
Будь-яка відповідь сервера завжди починається або з "+ОК" (команду користувача виконано правильно), або з "-ЕRR" (виникла помилка).  
Після завершення сеансу за допомогою команди "QUIT" здійснюється фіксація транзакції. Якщо з'єднання розривається з якихось зовнішніх причин до команди "QUIT", то здійснюється відкіт транзакції.  
Хоча протокол РОР3 і є простим у ньому, у ньому передбачено ефективний механізм захисту цілісності повідомлень - механізм транзакцій. Конфіденційність повідомлень електронної пошти під час їх передавання не забезпечують ані протокол РОР3, ані решта протоколів. Протокол РОР3 має серйозний недолік - примітивну схему автентифікації, що передбачає передавання паролю в відкритому вигляді. Перехоплений пароль може використати порушник для безконтрольного доступу до поштової скриньки користувача.

## ІМАР4

**ІМАР4** - новіший протокол читання електронної пошти. Протокол ІМАР4 підтримує такі операції:

* створення поштових скриньок;
* видалення поштових скриньок;
* перейменування поштових скриньок;
* перевірка надходження нових листів;
* видалення листів;
* встановлення і знімання прапорців операцій;
* пошук серед листів;
* читання обраних листів.

Протокол ІМАР4 не такий популярний як РОР3, через те, що в його основу покладено дещо іншу ідеологію. Фактично, протокол РОР3 передбачає, що на сервері всі листи знаходять в одному поштовому файлі. Протокол ІМАР4 призначено для можливості читати та обробляти пошту, яка зберігається на сервері, з клієнтських робочих станцій.   
Автентифікацію здійснює команда "LOGIN" (пароль передається у відкритому вигляді) або "AUTHENTICATE" (пароль передається зашифрованим). Сервер ІМАР4 за стандартом використовує порт 143.

### Загрози пов’язані з використанням електронної пошти

Існує два типи загроз пов'язаних з використанням електронної пошти. Загрози першого типу пов'язані з використанням сервісе електронної пошти. Як і будь-який інший сервіс, електронну пошту використовують не лише за призначенням, але й у зловмисних цілях. Переваги електронної пошти можуть легко обернутися на ризики. Умовно назвемо їх зловживанням електронною поштою.  
Загрози другого типу зумовлені недоліками у протоколах електронної пошти і помилками в реалізації програм, які забезпечують цей сервіс. Ці загрози зловмисники можуть реалізувати у вигляді атак, спрямованих як на поштовий сервер, так і на комп'ютер, де встановлено поштового клієнта. Назвемо їх атаками через систему електронної пошти.  
Доступність і дешевизна електронної пошти може стати її недоліком, позаяк порушники обирають саме цей сервіс для масового розсилання рекламних повідомлень. Легкість у використанні цього сервісу призводить до того, що його засовують некваліфіковані користувачі, які легко піддаються впливу так званої соціальної інженерії. Наявність можливості пересилати документи різних форматів створює загрозу поширенню вірусів та інших небезпечних програм. Електронну пошту можна також розсилати анонімно.  
Будь-яка з цих загроз може створити серйозні проблеми для користувачів, а особливо - для корпоративних користувачів (компаній). Це й зниження ефективності роботи, й втрата якості послуг інформаційних систем, й розкриття конфіденційної інформації.

### Пересилання електронною поштою шкідливих програм

Завдяки застосуванню МІМІ-стандарту за допомогою електронної пошти можна пересилати великі об’єми інформації різних форматів даних у вигляді вкладень (прикріплених до повідомлень файлів). Ця властивість зробила електронну пошту майже ідеальним середовищем для пересилання різних небезпечних вкладень. Якщо не впровадити належний контроль за використанням електронної пошти, це може призвести до дуже серйозних наслідків. Ефективним засобом є блокування певних типів файлів, до яких належать виконувані файли, бібліотеки, інсталяційні пакети, файли що можуть містити макроси, архіви. Блокування всіх зазначених типів вкладень суттєво обмежує функціональність електронної пошти, тому таку можливість слід використовувати обережно.

### Неконтрольоване використання електронної пошти

Великі бізнесові проблеми іноді виникають у компаніях через те, що співробітники використовують електронну пошту з метою, не пов'язаною з основною діяльністю (для обміну мультимедійним контентом, графічним, відео- і аудіофайлами, для приватного листування, ведення власного бізнесу з використанням поштових ресурсів компанії).

### Розсилання спаму

[**Спам**](http://uk.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B0%D0%BC) - це анонімна масова незапитана розсилка. Спам має три ознаки:

* Анонімність - приховується справжній виконавець розсилки;
* Масовість - саме масові розсилки належать до спаму, і лише вони є прибутковим злочинним бізнесом;
* Незапитаність - рекламні розсилки можуть бути небажаними, але якщо користувач сам на них підписався, то такі розсилки не можна вважати спамом.

Захист від спаму полягає у впровадженні певних політик оброблення поштових повідомлень, які здебільшого передбачають перегляд вмісту повідомлень і здійснення фільтрації за визначеними правилами. Є два базових підходи до фільтрації:

* формальний;
* семантичний.

Формальний підхід передбачає фільтрацію за списками і за формальними ознаками повідомлення. Використовуються так звані чорні і білі списки. Чорний список - це список адрес, які вважаються спамерскими. Є організації, які складають і розповсюджують за підпискою такі списки. Білий список - це список адрес, пошту з яких слід приймати в будь-якому разі. До формальних ознак повідомлення належать особливості полів у його заголовку (відсутність адреси відправника, велика кількість адресатів, некоректні технічні заголовки), а в деяких випадках і формат самого листа.  
Семантичний підхід полягає у проведенні аналізу вмісту листів і їх фільтрації за сигнатурами або за лінгвістичними евристиками. Сигнатури дозволяють надійно розпізнавати вже відомі спамові повідомлення. Евристики - це набори характерних словосполучень з урахуванням їхніх ймовірнісних характеристик. Евристики дозволять з певною мірою надійності розпізнавати нові, ще не відомі спамові повідомлення.

### Анонімне відсилання електронної пошти

Більшість зловживань електронною поштою не були б так поширені, якби порушника можна було виявити і покарати.   
В інтернеті є сервери, які надають послуги з анонімного доступу до ресурсів мережі. До таких належать так звані анонімайзери та проксі-сервери. З одного боку, вони справді приховують адресу користувача, з іншого - більшість з них усе зберігає Ір-адресу в журналі реєстрації. Таким чином, якщо за допомогою такого сервера було здійснено злочинні дії, то адресу, з якої ініційовано такі дії легко виявити.   
Справжні зловмисники використовують ланцюги серверів, оскільки, прослідкувати довгу послідовність серверів, на кожному з яких адресу було приховано, дуже складно. Зловмисники пишуть власні програми, які здійснюють формування повідомлень. Такі програми розміщують на серверах, що дають змогу розміщувати і запускати на виконання власні програми, або на серверах, скомпрометованих внаслідок атаки.  
Анонімне відправлення електронної пошти цілком можливе, але є нелегкою справою. Підробку, як правило, можна помітити, але визначити справжнього відправника складно.

## **Apache James**

**Apache James,** відомий **Java Apache Пошта Enterprise Server** або будь-якої їх варіації, є [відкритим вихідним кодом](https://translate.googleusercontent.com/translate_c?depth=1&hl=uk&ie=UTF8&prev=_t&rurl=translate.google.com&sl=en&tl=uk&u=http://en.wikipedia.org/wiki/Open_source&usg=ALkJrhik3Zcul4xzM5LsPyeFLOqHyPRfOw) [SMTP](https://translate.googleusercontent.com/translate_c?depth=1&hl=uk&ie=UTF8&prev=_t&rurl=translate.google.com&sl=en&tl=uk&u=http://en.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol&usg=ALkJrhhb2guXFvzRPRlnZsHEkGVICZ7N8w) і [POP3](https://translate.googleusercontent.com/translate_c?depth=1&hl=uk&ie=UTF8&prev=_t&rurl=translate.google.com&sl=en&tl=uk&u=http://en.wikipedia.org/wiki/Post_Office_Protocol&usg=ALkJrhiz2LsNxnpLyIlhiP-9W6cRPVZfMw) [поштовий агент](https://translate.googleusercontent.com/translate_c?depth=1&hl=uk&ie=UTF8&prev=_t&rurl=translate.google.com&sl=en&tl=uk&u=http://en.wikipedia.org/wiki/Mail_transfer_agent&usg=ALkJrhixT-3h9GqHazdO5xM1ttjQkECwrw) і [NNTP](https://translate.googleusercontent.com/translate_c?depth=1&hl=uk&ie=UTF8&prev=_t&rurl=translate.google.com&sl=en&tl=uk&u=http://en.wikipedia.org/wiki/Network_News_Transfer_Protocol&usg=ALkJrhicb1-u0h47gGQ3WSwI5MRTd37S-w) [сервер новин](https://translate.googleusercontent.com/translate_c?depth=1&hl=uk&ie=UTF8&prev=_t&rurl=translate.google.com&sl=en&tl=uk&u=http://en.wikipedia.org/wiki/News_server&usg=ALkJrhg-PmNW1zurwMCCkpy_ESWZfPaXXA) , повністю написаний на [Java](https://translate.googleusercontent.com/translate_c?depth=1&hl=uk&ie=UTF8&prev=_t&rurl=translate.google.com&sl=en&tl=uk&u=http://en.wikipedia.org/wiki/Java_(programming_language)&usg=ALkJrhi8uSA0wLDklr0Ey6rDokoeFp-wIQ) . James підтримується [Apache Software Foundation](https://translate.googleusercontent.com/translate_c?depth=1&hl=uk&ie=UTF8&prev=_t&rurl=translate.google.com&sl=en&tl=uk&u=http://en.wikipedia.org/wiki/Apache_Software_Foundation&usg=ALkJrhhu5PD5fioZkT5BSIjFo_61MHQHfQ) , з початкових внесків від [Serge Knystautas](http://en.wikipedia.org/w/index.php?title=Serge_Knystautas&action=edit&redlink=1). [IMAP](https://translate.googleusercontent.com/translate_c?depth=1&hl=uk&ie=UTF8&prev=_t&rurl=translate.google.com&sl=en&tl=uk&u=http://en.wikipedia.org/wiki/Internet_Message_Access_Protocol&usg=ALkJrhh_9E4tOqj7-zA8TZoLxPwL5sfuXQ) підтримка була додана станом на попередній версії 3.0-M2, який в даний час вимагає Java 1.5 або більш пізньої версії.

The James project manages the Apache Mailet [API](https://translate.googleusercontent.com/translate_c?depth=1&hl=uk&ie=UTF8&prev=_t&rurl=translate.google.com&sl=en&tl=uk&u=http://en.wikipedia.org/wiki/Application_programming_interface&usg=ALkJrhhW44WxrlOCwe-xqM63wzMNuM33_Q) which defines "matchers" and "mailets". Проект James управляє Apache Mailet [API](https://translate.googleusercontent.com/translate_c?depth=1&hl=uk&ie=UTF8&prev=_t&rurl=translate.google.com&sl=en&tl=uk&u=http://en.wikipedia.org/wiki/Application_programming_interface&usg=ALkJrhhW44WxrlOCwe-xqM63wzMNuM33_Q) , який визначає "обчислювачів" і "mailets". These allow users to write their own mail-handling code, such as to update a database, build a message archive, or filter [spam](https://translate.googleusercontent.com/translate_c?depth=1&hl=uk&ie=UTF8&prev=_t&rurl=translate.google.com&sl=en&tl=uk&u=http://en.wikipedia.org/wiki/E-mail_spam&usg=ALkJrhjowxcJj8jr-6tvSR_s0pi2EeVSBg) . [[ 1 ]](https://translate.googleusercontent.com/translate_c?depth=1&hl=uk&ie=UTF8&prev=_t&rurl=translate.google.com&sl=en&tl=uk&u=http://en.wikipedia.org/wiki/Apache_James&usg=ALkJrhjoZV1YEnVwo56wxse-OZkrNv8Xwg#cite_note-overview-1) A matcher is used to classify messages based on some criteria, and then determines whether the message should be passed to an appropriate mailet for processing. Вони дозволяють користувачам писати власний код поштою з управління, наприклад, для оновлення бази даних, створювати архів повідомлень, або фільтр [спам](https://translate.googleusercontent.com/translate_c?depth=1&hl=uk&ie=UTF8&prev=_t&rurl=translate.google.com&sl=en&tl=uk&u=http://en.wikipedia.org/wiki/E-mail_spam&usg=ALkJrhjowxcJj8jr-6tvSR_s0pi2EeVSBg) . [[1]](https://translate.googleusercontent.com/translate_c?depth=1&hl=uk&ie=UTF8&prev=_t&rurl=translate.google.com&sl=en&tl=uk&u=http://en.wikipedia.org/wiki/Apache_James&usg=ALkJrhjoZV1YEnVwo56wxse-OZkrNv8Xwg#cite_note-overview-1) збігів використовується для класифікації повідомлень на основі деяких критеріїв, а потім визначає, чи є повідомлення повинно передаватися на відповідний mailet для обробки. Mailets are so-called due to their conceptual similarity to a [servlet](https://translate.googleusercontent.com/translate_c?depth=1&hl=uk&ie=UTF8&prev=_t&rurl=translate.google.com&sl=en&tl=uk&u=http://en.wikipedia.org/wiki/Servlet&usg=ALkJrhiiwIORoEIRUG0f-IA4xBC1bCfAcQ) , [[ 5 ]](https://translate.googleusercontent.com/translate_c?depth=1&hl=uk&ie=UTF8&prev=_t&rurl=translate.google.com&sl=en&tl=uk&u=http://en.wikipedia.org/wiki/Apache_James&usg=ALkJrhjoZV1YEnVwo56wxse-OZkrNv8Xwg#cite_note-ibm-5) and arose because [Sun Microsystems](https://translate.googleusercontent.com/translate_c?depth=1&hl=uk&ie=UTF8&prev=_t&rurl=translate.google.com&sl=en&tl=uk&u=http://en.wikipedia.org/wiki/Sun_Microsystems&usg=ALkJrhgclIPAgoJdpm0_egxGcmU_Lkxt3w) declined a proposal to include mail-handling in the servlet implementation. [[ 3 ]](https://translate.googleusercontent.com/translate_c?depth=1&hl=uk&ie=UTF8&prev=_t&rurl=translate.google.com&sl=en&tl=uk&u=http://en.wikipedia.org/wiki/Apache_James&usg=ALkJrhjoZV1YEnVwo56wxse-OZkrNv8Xwg#cite_note-boardminutes-3) James ships with a variety of pre-written matchers and mailets to serve common tasks. [[ 5 ]](https://translate.googleusercontent.com/translate_c?depth=1&hl=uk&ie=UTF8&prev=_t&rurl=translate.google.com&sl=en&tl=uk&u=http://en.wikipedia.org/wiki/Apache_James&usg=ALkJrhjoZV1YEnVwo56wxse-OZkrNv8Xwg#cite_note-ibm-5) Many sets of mailets and matchers can be combined to produce sophisticated and complex functional behaviour. Mailets так звані через їх концептуального схожості з [сервлет](https://translate.googleusercontent.com/translate_c?depth=1&hl=uk&ie=UTF8&prev=_t&rurl=translate.google.com&sl=en&tl=uk&u=http://en.wikipedia.org/wiki/Servlet&usg=ALkJrhiiwIORoEIRUG0f-IA4xBC1bCfAcQ) , і, вставши з-за [Sun Microsystems](https://translate.googleusercontent.com/translate_c?depth=1&hl=uk&ie=UTF8&prev=_t&rurl=translate.google.com&sl=en&tl=uk&u=http://en.wikipedia.org/wiki/Sun_Microsystems&usg=ALkJrhgclIPAgoJdpm0_egxGcmU_Lkxt3w) відхилив пропозицію про включення пошту обробки в реалізації сервлета. James кораблі з різними попередньої письмової обчислювачів і mailets щоб служити в спільних завдань. Безліч наборів mailets і обчислювачів можуть бути об'єднані, щоб справити складний і складний функціонального поведінки.

The Apache James project also produces pure Java libraries for implementing Sender Policy Framework (SPF), the Sieve mail filtering language, and parsing MIME content streams, independent of Sun's JavaMail API. Проект Apache James також виробляє чисті Java бібліотеки для реалізації Sender Policy Framework (SPF), фільтрації пошти, і аналіз потоків контенту MIME, незалежно від JavaMail API.

## Технологія

|  |  |
| --- | --- |
| Назва вирішення (технології) | ApacheJames |
| Сфера використання (яку проблему вирішує технологія) | SMTP, POP, IMAP |
| Обмеження по використанню |  |
| Обов”язкові компоненти вирішення (апаратні, програмні, структурні) |  |
| Протоколи що використовуються | SMTP, POP, IMAP |
| Інші вирішення (технології) що вирішують подібну задачу | Microsoft Exchange, hMail Server, Eudora Internet Mail Server |
| Переваги технології | * Система з відкритим кодом * Проект верхнього рівня Apache. * Має комерційну підтримку * Підтримує стандартні протоколи * Інтегрується з LDAP * Webmail як суб проєкт * Добре розширюється, має Java API * Легкий в інсталяції |
| Недоліки технології | * Не активна розробка * Немає підтримки Active Sync, Gmail IMAP Extensions, OAuth * Адміністрування тільки через командний рядок |

## Реалізація



Рис . Діаграма рішення

Основа рішення поштовий сервер Apache James 3.0. Налаштування:

|  |  |  |
| --- | --- | --- |
| Назва | Значення | Комантар |
| Domain name | chaliy.local |  |
| Протоколи | SMTP, POP3, IMAP | Захищенні налаштування, підтримка авторизації і TLS |
| Базаданних | PostgreSQL |  |
| Платформа | Oracle Java 1.6 |  |
| Контейнер | Docker |  |
| Аутентифікація | LDAP | Використовується OpenLDAP |
| Інфраструткруа ключів | OpenPGP | PGP має свої недоліки, в перспективі має використовуватись S/MIME інфраструктура |

Для доступу через веб браузер використовується Apache James Hupa:

|  |  |  |
| --- | --- | --- |
| Назва | Значення | Комантар |
| Протоколи | HTTP | Захищенні налаштування, SSL Only |
| Платформа | Oracle Java 1.6 |  |
| Контейнер | Docker |  |
| Аутентифікація | LDAP | Використовується OpenLDAP |
| Веб сервер | Apache Tomcat |  |

Для керування користувачами використовується OpenLDAP:

|  |  |  |
| --- | --- | --- |
| Назва | Значення | Комантар |
| Domain name | chaliy.local |  |
| Протоколи | LDAP | TLS |
| Базаданних | PostgreSQL |  |
| Платформа | Oracle Java 1.6 |  |
| Контейнер | Docker |  |
| Клієнт | Apache Directory Studio | Базуєтсья на Eclipse |

## Висновки

Дослідив технологію Apache James та спроектував рішення в галузі КМ на базі цієї технології.