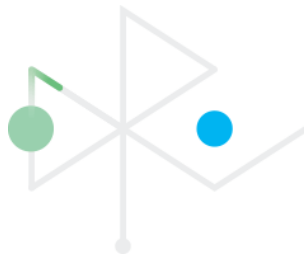
Introduction to Kopano

Jelle van der Waa

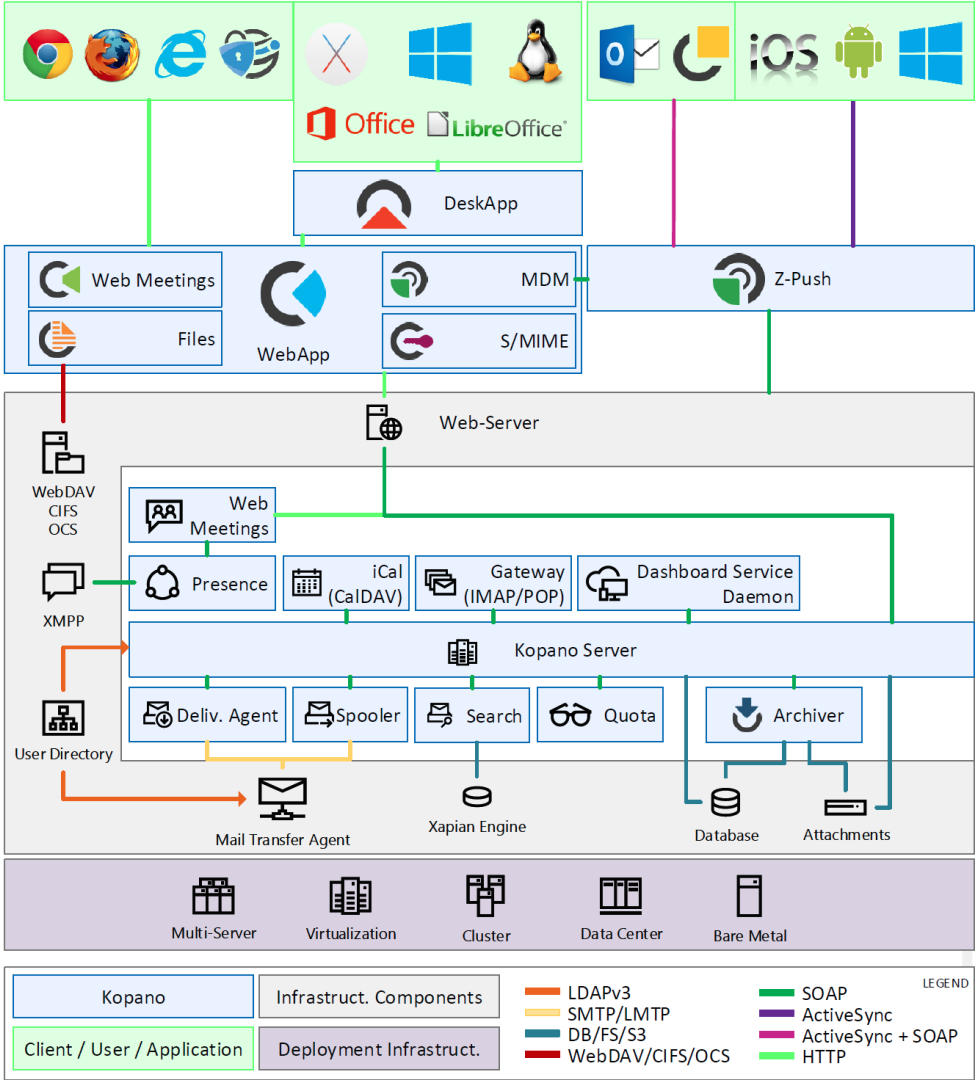
Introduction

- AGPLv3
- Fork of Zarafa
- Implements the MAPI Specification
- MySQL for MAPI data storage
- Attachments on disk or S3 based (protocol)
- Core consists of C++, Python, PHP

https://www.kopano.io
https://stash.kopano.io/repos?visibility=public

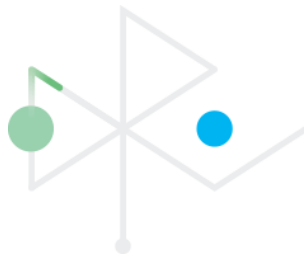


Architecture



Architecture

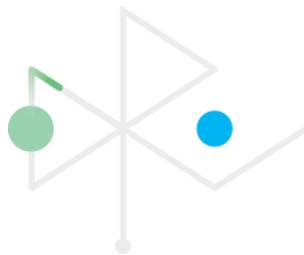
- All components use SOAP for transport
- Provide bindings for PHP, Python for MAPI clients
- Use of any MTA is possible (postfix recommended)
- All data is stored as MAPI objects
- Does not implement the DCE/RPC, MAPI/HTTP or “Outlook Anywhere”
- Multiple user backends, DB / LDAP / AD / PAM
- Python plugin framework for sending/receiving mails



Z-Push

- Open Source ActiveSync implementation
- Written in PHP (uses php-mapi extension)
- Supports multiple backends (Kopano, Maildir, Vcard, IMAP)
- Kopano Outlook Extension (KOE) – Extends the featureset of of OL's ActiveSync Implementation

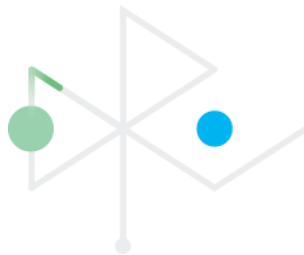
http://z-push.org
https://stash.z-hub.io/projects/ZP/repos/z-push/browse



Downstream

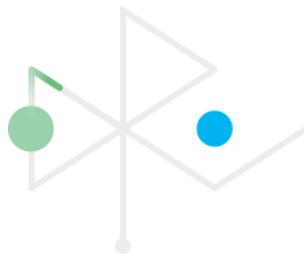
- Debian – Process of getting into Debian stretch
  https://wiki.debian.org/Groupware/Kopano

- OpenSUSE – Build Service repository
  https://build.opensuse.org/project/show/server:mail:kopano



Python-kopano

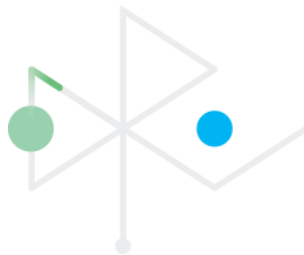
- 2 years ago started hacking on a Python high level API
- Search / backup and debugging tools build on top of it
- Easy interface to automate / extend Kopano (from a client)



Python-kopano

```
>> import kopano
>> user = kopano.User('user1')
>> # Send email
>> user.inbox.outbox.create_item(subject='hello', body='empty body!',
to='j.vanderwaa@Kopano.com').send()

>> # access MAPI property
>> item = user.inbox.items().next()
>> print item.prop(PR_SUBJECT), item.subject
>> item.subject = 'new subject'
```



Questions