

Tecnologia

Licenza software

Applicazione disaccoppiata

Parte server: installazione on premise

Parte server

Parte frontend

Parte db

docker

Conflitti con librerie

Personalizzabile con plugin

Vulnerabilità note

Valutazione del carico

Livello di sicurezza

Interfaccia

Usabilità

Possibilità di utilizzo su mobile

Accattivante

Dimensione community

Collaboratività

Backup dei dati

Chatbot

Esistenza di un marketplace o plugin



Per zulip e mattermost si potrebbero creare degli slash commadn che permettono di fare azioni personalizzate

Zulip (<https://zulip.readthedocs.io/en/latest/overview/architecture-overview.html>)

Apache 2.0

Ni parte "server" e frontend web costituiscono un monorepo, ci sono le viste python che in realtà sono api. Un sacco di templating diversi per il front end e il backend

Sì → su openSuse possiamo installarlo? Il supporto è per debian e ubuntu (nota ubuntu 18.08?? sicura robbly?)

Python – Django + Tornado (dentro c'è anche Rabbit) Python 3.3 / 3.6 con tipi statici. Redis e MemCached (caching mod time server LONG POLLING + nginx

Usa libreria zulip-js ok per typescript e eslint. Dubbio su babel (usa plotly? La versione più indietro ma non dovrebbe essere javascript che si chiama handlebars. La libreria zulip-js è il collegamento per le api, servono delle chiavi username e realm

Postgresql 9.1+ con gestione delle migration

solo sperimentale

BARRETTE

importante è avere webhook per spedire sui canali – non è spiegato bene come buildare e modificare il frontend, ma in ge

Le vulnerabilità lato fe sono legate alle librerie utilizzate, tipicamente XSS (cross site scripting)

il problema è legato a python, non penso tornado risolva, occorre molta memoria perchè le cache che usa sono tutte in m LDAP/Active directory integration, comunicazioni TLS e SSH anche on premise, possibilità di crittare i dati. Zulip gira con utente parti tipo sentry che spediscono dati relativi agli errori (logging e monitoraggio in tempo reale)

solo chat di piccoli gruppi o messaggi privati, con webhook verso i principali integration software (altassian, bitbucket), in (es zoom) thumbor come gestore di immagini. Non ho capito cosa è uno stream. Le conference call si fanno su jitsi esiste l'app ma importa la distribuzione su dispositivi protetti. L'interfaccia web è da preferire a me l'interfaccia sembra degli anni 2000, poco moderna, sembrano un sacco di librerie datate

Dichiara di essere la più grande di tutte

possibilità di vedere chat organizzate. Non supporta altre funzionalità se non la chat o tramite integrazione di servizi tipo z più che un backup è la nuova piattaforma che dovrebbe fornire un plugin di importazione. Zulip supporta da mattermost no, sono in beta i BOT interattivi, interagiscono ai messaggi di zulip e fanno cose, si basano sulle api ma devi avere un'ap sì, ma è rivolto alle integrazioni verso i tool di continuous integration / grafana / sistemi di monitoraggio. Ma alcuni hanno u guida che spiega cosa fare <https://zulip.readthedocs.io/en/latest/production/upgrade-or-modify.html#modifying-zulip>. Ma python non tanto la parte di frontend.

Note: zulip si contrappone a slack nella sua presentazione sconsigliata l'interfaccia web, consigliano la desktop

Dietro c'è DROPBOX

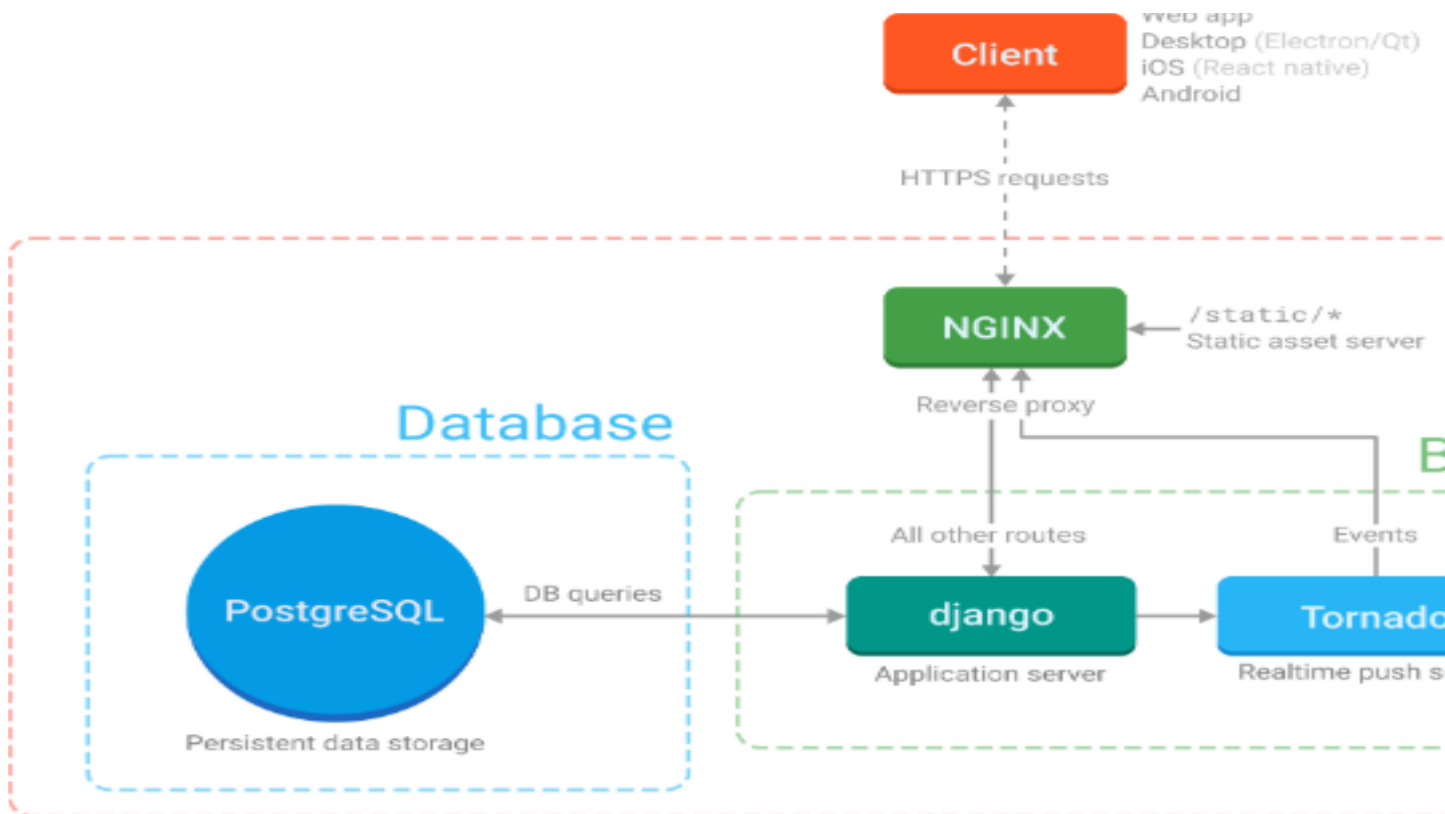
nominato spesso integrazione con Zapier

Potrebbe essere configurato da zabbix verso zulip per avvisare di qualcosa?

Sì → <https://zulipchat.com/integrations/doc/zabbix>

Cronograph → non pervenuto

Il sistema di notifica di malfunzionamento nativo è nagios



Allora, il sistema di spedizione delle notifiche è dato da un payload che rappresenta l'evento e gli id dei client che devono ricevere la notifica. Il payload viene inviato a rabbitmq e rabbitmq poi lo passa a tornado che risponde al client.

Il paradigma di funzionamento è il long polling, cioè il client fa la chiamata al server ma non riceve una sola risposta "consigliata", viene tenuto aperto dal server che continua a mandare dati quando li ha aggiornati.

Dicono che le websocket hanno problemi sui tipi di client

Praticamente mantiene una coda di eventi per client connesso

Tornado non accede al db nè ad altre persistenze, è il sistema che tiene su i collegamenti long polling.

i collegamenti al db sono fatti tramite django

Le connessioni client / server sono stoppate dopo 10 minuti di inattività, c'è una politica di recovery per la failure di una connessione.

Si evitano troppe query al db per tirare su lo stato iniziale di una connessione.

un client si connette e descrive le sue subscription

Django chiama tornado in http e gli chiede di creare una coda, poi fa il recupero dati e gli manda quello che serve a tornado

Grande uso di caching -> IN MEMORIA era grande consumo di memoria

tipo apro una issue

Mattermost

MIT

Sì, ma poi vengono gestite sotto lo stesso dominio, quindi sembra un tutt'uno.

Sì, ci sono anche i pacchetti per SUSE (Leap e anche altre denominazioni)

Golang, NGINX sentry elastic search Bleeve come search engine

React + parecchio javascript

MySQL o postgres

sì

non nel frontend, tuttavia usa una versione dedicata di redux che si chiama mattermost redux che non è distribuita da n...
da github. Non è bellissimo

Sezione dedicata all'embedding con widget js, l'estensione si fa tramite plugin (nella team edition) in modo da non toccare il
sistema si chiama Plugin framework. IL client frontend

C'è un bulletin periodico che avvisa delle possibili vulnerabilità

Il più sicuro, poichè utilizzato in molte installazioni onpremise, ma LDAP/Active directory sono nella edizione enterprise

fa quello che serve meno semplice di zulip

sì ma tanto non possiamo distribuire

Sì,

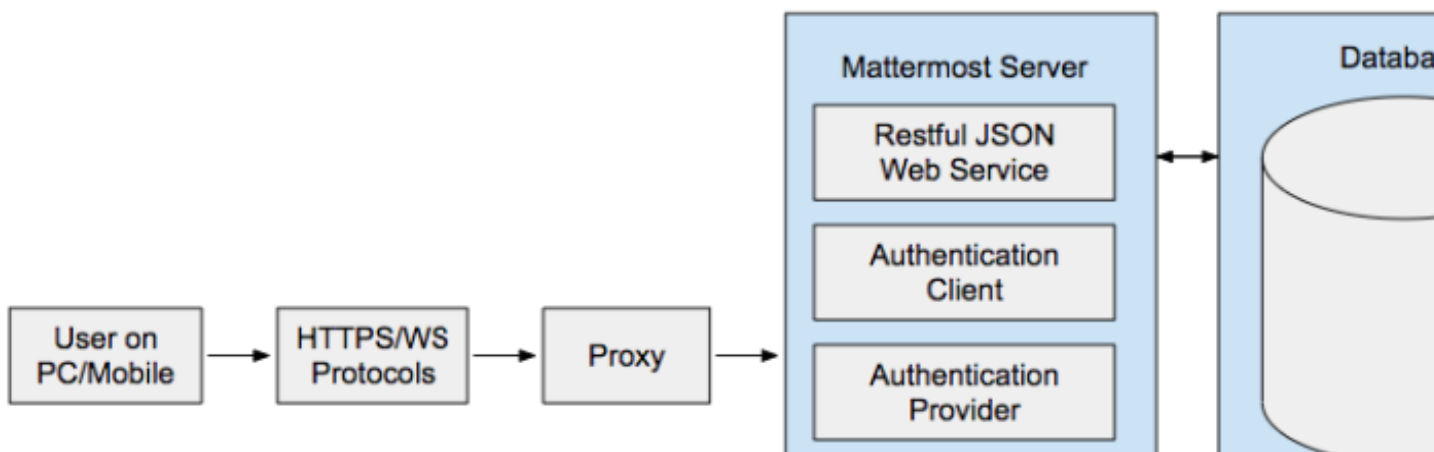
grande crescita nel 2019

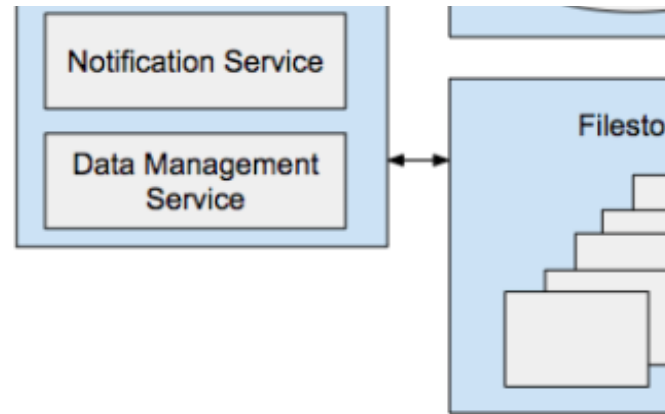
sì dedicato al remote working

si seguono le solite procedure di backup

s' <https://mattermost.com/blog/live-chat-using-mattermost/>

sì, <https://integrations.mattermost.com/>





Implementa http(s) e web socket

Alta affidabilità solo nella versione enterprise

Cloud oriented

Qui il notification service funziona prevalentemente verso android e ios

virtual office -

virtual campus -> enterprise

Attenzione! Push notification è significativo per le app e si appoggia sui sistemi google e attenzione, arrivano sul "dispositivo" non necessariamente il browser

Sembra che al suo interno ci sia uno schedatore di Job

skype for business plugin

Riot

Apache 2.0

Sì

sì, homeserver si chiama synapse (opensuse)

Python > 3.5 twisted per la parte real time – Redis (perchè?) (ocio usa sentry) NGIX

React+jS basato su sdk matrix-react-sdk con libreria per accedere alle api

SQLLite postgres

Sì (ma non sembra parte della distribuzione ufficiale)

la parte web è riot web, è previsto che possa girare sotto docker, basato su

Idap comunicazioni crittate, il sistema è decentralizzato tLS https

si basa sul concetto di stanza, si partecipa a delle stanze. Server federati vedono le stesse stanze, usa websocket. Jitsi è la conference call. E' possibile un sistema interno? Al momento è esterno

Moderna

fuori dalla beta dal 2019

dedicato e orientato al concetto di comunicazione sicura

backup dei dati legati al db

no

sì, i collegamenti con altre applicazioni si chiamano bridge

no single point of failure (ma se il server è uno solo)

è dato per grandi moli di utenti

integrazione con go e sdk per utilizzarlo in javascript

instant messaging voip decentralizzato

Zulip is an emerging leader of business collaboration tools. Known for its killer email threading model, Zulip assures its user to efficiently catch up on the important notifications and conversations while muting the irrelevant ones. Zulip has made a mark among great many collaboration applications and known for providing the advantages of real-time chat. No other application is known for handling synchronous and asynchronous communication as intuitively as Zulip does.

Mattermost, as open-source messaging software, has earned the trust of millions of users. Not only does it allow the users to create an intuitive workspace but also collaborate across teams without worrying about data privacy or security. A great number of high-security enterprises have been working through Mattermost by connecting their team and deploying automation for maximum benefit.

<https://www.tro>

<https://opensou>

<https://resources.whitesourcesoftware.com/blog-whitesource/top-10-apache-license-questions-answe>

red#:~:text=What%20is%20the%20difference%20between%20Apache%20License%202.0%20and%

MIT%3F&text=Basically%2C%20you%20can%20do%20whatever,is%20also%20a%20permissive'

%20license .