Socket : یک کتاب خانه سطح پایین برای ارتیاطات شبکه‌ای که با آن میتوان در شبکه درخواست داد و آنها را کنترل کرد.

لینک داکیومنت socket : <https://docs.python.org/3/library/socket.html>

مزایا :

1. قابلیت کنترل کامل شبکه
2. تقریبا برای هر کاری در شبکه قابل استفاده است
3. از ماژول های داخلی پاتون است و نیازی به نصب شدن ندارد

معایب:

1. نیازمند پیاده سازی تقریبا تمامی بخش های برنامه به صورت دستی
2. به طور پیش فرض عملیات های این ماژول قابلیت بلاک کردن دارند اگرچه حالت های بدون بلاک کردن نیز دارد

Zeromq ( pyzmq ) : یک کتاب خانه سطح بالا نا‌همگام برای مکالمه در شبکه

لینک داکیومنت zmq : <https://pyzmq.readthedocs.io/en/latest>

مزایا :

1. ساخته شده برای مکالمه های سریع در شبکه
2. پشتیبانی از حلات های مختلف برای مکالمه مانند انتشار/ دنبال کردن و درخواست / پاسخ و ...
3. بدون بلاک کردن، پشتیبانی از برنامه نویسی ناهمگام

معایب :

1. قابلیت های پیشرفته آن و روش های ارتباطی آن سختی کار با آن را افزایش میدهد
2. پیشرفتگی آن برای کار های ساده زیاد است

Celery : کتاب خانه ای ناهمگام براساس توزیع پیام های ارسالی

لینک داکیومنت : <https://docs.celeryproject.org/en/stable>

مزایا :

1. ایده‌آل برای مدریت کار های ناهمگام
2. از انواع واسط های پیام پشتیبانی میکند مانند rabbitMQ, redis
3. قابلیت کنترل کردن تعداد زیاد عملیات

معایب :

1. پیچیدگی زیاد در راه اندازی
2. برای کار های سبک و کوچک بیش از حد پیچیده است

rabbitMQ ( pika ) : کتاب خانه‌ی سمت کلاینت برای rabbitMQ و یک واسط پیام رسانی معروف

لینک داکیومنت : <https://pika.readthedocs.io/en/latest>

مزایا :

1. دارای صف قوی ارسال پیام با قابلیت هایی مانند پایداری و تصدیق و تلاش دوباره
2. پشتیبانی از الگو های پیام رسانی مختلف و گزینه های مسیریابی مانند direct ,topic ,fanout و تعویض header
3. یک گزینه محبوب برای واسط گری پیام با کامیونیتی بزرگ و داکیومنت بزرگ

معایب :

1. نیازمند ساخت و نگه داری از یک سرور rabbitMQ است که کار را پیچیده تر میکند
2. برای بخش سرور نیاز به یادگیری و راه اندازی rabbitMQ است و برای بخش کلاینت نیاز به یادگیری و راه اندازی pika است که پیچیدگی کار را بسیار زیاد میکند

Kafka (confluent-kafka ) : بخش کلاینت آپاچی کافکا

لینک داکیومنت : <https://docs.confluent.io/platform/current/clients/python.html>

مزایا :

1. ساخته شده برای حجم های زیاد داده با تاخیر پایین
2. با توزیع داده توسط واسط های پیام رسان از خطای یاد جلوگیری میکند
3. دارای ابزار های زیاد برای پردازش داده و مدیریت جریان داده

معایب :

1. پیاده سازی و کانفیگ کردن کافکا میتواند پیچیده و نیازمند مدیریت کامل واسط های پیام رسان و تاپیک ها و پارتیشن ها باشد
2. برای اجرای کافکا نیازمند سیستم اساسی و نگه داری از آن است و برای کار های ساده پیچیده است

Twisted : یک موتور شبکه براساس رویداد ها که از پروتکل های مختلف پشتیبانی میکند

لینک داکیومنت : <https://twistedmatrix.com/trac/>

مزایا :

1. از برنامه نویسی ناهمگام پشتیبانی میکند
2. از پروتکل های مختلف شبکه پشتیبانی میکند
3. برای پروژه های سطح بالا مناسب است

معایب :

1. ساختار رویدادی و ناهمگام میتواند پیچیده باشد
2. برای کار های ساده میتواند زیادی پیچیده باشد

**برای انتقال داده بین کامپیوترها بهترین انتخاب به نیاز خاص شما بستگی دارد مانند عملکرد، مقیاس پذیری، پیچیدگی و نوع داده ها. خلاصه تهیه شده توسط "chatgpt4-o mini" :**

1. برای عملکرد بالا و جریان داده لحظه ای :
2. Apache kafka
3. ZeroMQ
4. برای پیام رسانی قابل اطمینان و منعطف همراه با مدیریت کار ها :
5. rabbitMQ ( pika )
6. celery
7. برای ارتباط ناهمگام و http api های مدرن :
8. Httpx
9. برای کنترل سطح پایین و پروتکل های دست ساز :
10. Socket