After Disccusion i came to conclusion that REST API's is a better solution for SoMe Application and i have give my Arguments and Reasoning below:

Arguments:

- **Resource-Based Design:** REST organizes data into discrete resources; locating and manipulating social media entities-a set of well-bounded and therefore well-known entities such as users or posts-is relatively easy.
- **Standardization and Familiarity:** The REST APIs make use of conventions in HTTP that are widely recognized, such as GET, POST, etc., which already familiarize most developers. It therefore takes a shorter period to learn and is easier to work with.
- **Stateless for Scalability:** In REST, every request is self-sufficient; the server does not maintain any previous context from the client. This makes scaling quite easier because servers can handle numerous independent requests in parallel.
- **Caching for Performance:** REST provides an opening for caching to be applied, by which frequently requested data-for instance, profiles or posts-can be cached either locally or on intermediaries. This will reduce response times and lower server load.
- **Easy Versioning and Maintenance:** REST allows versions in the URL itself, say, /v1/users, which can be updated separately without affecting the existing clients. This is useful in iterating social media platforms.
- **Scalable to High Traffic:** The design of REST allows distribution requests across multiple servers to handle the high volumes typical for social media applications.
- **Improved Security Compatibility:** REST is compatible with HTTPS, which allows encryption of data; OAuth for limited but secure access to the users' private information, thereby protecting against disclosure and assuring safety in handling user data.

Reasoning:

- **Resource-Based Design:** Organizing REST around resources-say, users, posts, comments-there's an intuitive match between the design of the system and what's on offer on social media. This makes the API not only more workable for the developer but more usable in itself.
- **Standardization and Familiarity:** REST APIs use standardized HTTP methods like GET, POST, etc., which are quite familiar to developers. This increases development speed and decreases onboarding time significantly.

- **Stateless for Scalability:** REST is stateless; each request is independent, and thus not reliant on information or data held by the server. This independence enables better scaling and easier load balancing, which makes a significant effect on social media applications with high traffic.
- **Caching for Performance:** REST allows HTTP caching of resources frequently accessed, such as user profiles or images, to be stored locally and retrieved quickly rather than continuously requested from the server, thereby easing server load and improving the user experience.
- **Versioning and Maintenance:** The versioning in REST is straightforward, using the URL itself-for example, /v1/users. It means that updates are possible without affecting existing applications, which is crucial in evolving social media platforms that have to support older releases of their code.
- **Scalable for High Traffic:** Its stateless nature and the ability to scale out by adding server clusters enable REST to handle several million requests, perfect for high volume applications.
- **Improved security compatibility:** REST is compatible with HTTPS and OAuth; these allow encrypted communication and secure, fine-grained access to user information-the foundation of privacy for social networking.