

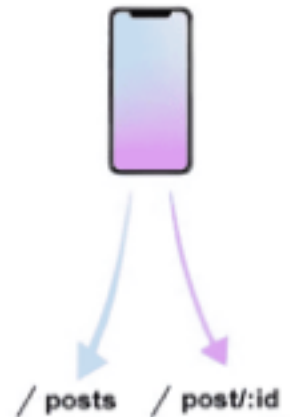
CHOISIR LE BON OUTIL

| Temps réel VS résilience | Client - server : request / response | Server - clients : event broker | Peer-to-peer | Batch |
|--------------------------------|---|------------------------------------|-------------------------------|---|
| TR > résilience | REST/JSON GraphQL gRPC | Websocket | webRTC ipfs scuttlebutt | <i>BAD IDEA</i> |
| Résilience > TR | <i>RPC sur Message Queue</i> | Message Queue (RabbitMQ) | zeroMQ | <i>Timer process</i> |
| Pas de TR | <i>BAD IDEA</i> | Agent de transfert (Fluentbit) | Blockchain | ETL (Talend) Data pipeline (Fluentd) |
| IPC (no networking) | <i>RPC sur Unix Socket</i> | Unix Socket | Named pipe (FIFO) | crontab +.sh |

WEBSERVICES

REST/JSON LE STANDARD DES ANNÉES 2010'S

{...} REST



REST = URI + VERBE + TYPE de média (*application/JSON, application/XML, ...*)

Pas de typage des ressources en standard mais possible via Json Schema ou Json-LD + schema.org

REST « RPC » ... a des problèmes :

- Adherence back-front
- Empêche la mise en cache

REST « HATEOAS » ... a des problèmes :

- Orienté « ressources » quand les bonnes pratiques poussent à l'orienté « domaine »
- Problème de N+1 requêtes