Replication for Configuration Data

Greg Brail

Situation

Our business (on the Apigee team at Google) is API management

- Managing, securing, and controlling our customers’ API endpoints

We want to put “message processing agents” on customers’ networks

- Agents run on typical corporate networks with all the restrictions

Only recently joined Google

- Our solution has to work across clouds, data centers, and networks

Observations

Lots of our data is read-mostly

Configurations of APIs, applications, products, and the like

Deployments of application-like things (we call them “proxies”)

Data is not huge -- 100,000s of rows for the largest customer

Data is consumed at same rate as API calls (lots)

Data changes slowly (a few times / minute)

Desired Solution

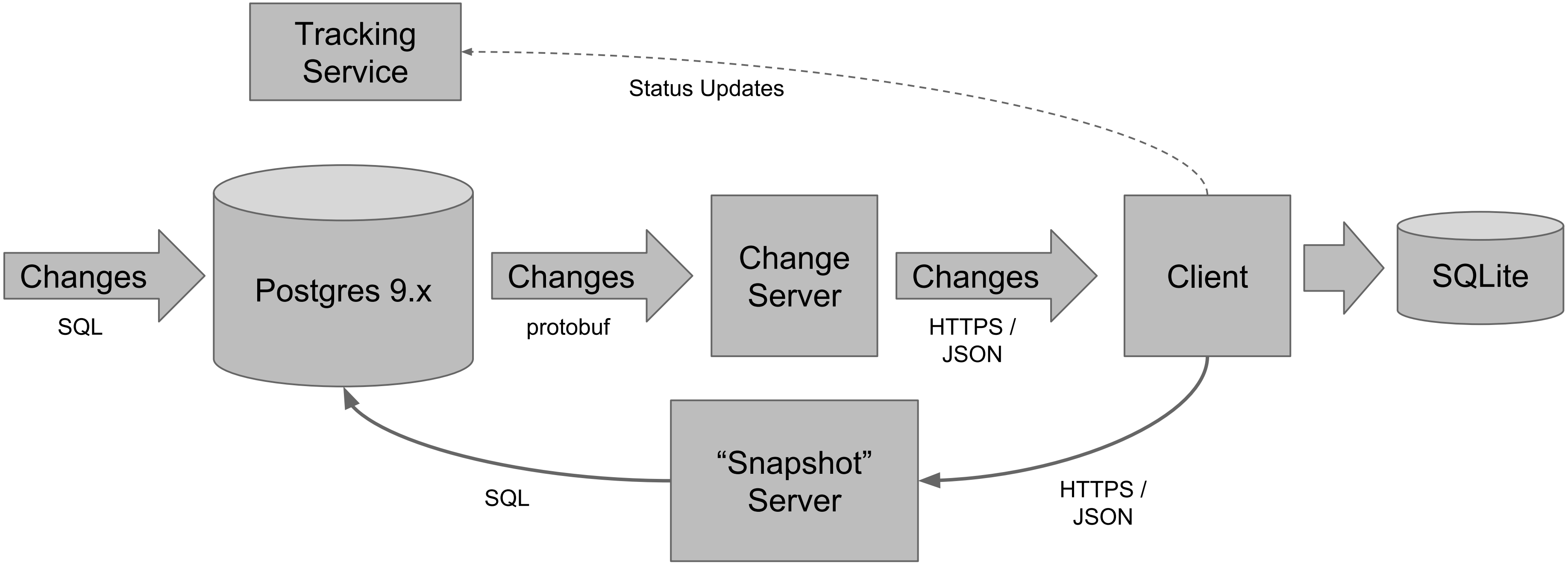
Put the data on the endpoints

Get it there in an eventually-consistent way

Solution

1. Use Postgres to store state centrally
2. Use Postgres “logical replication” to capture changes in consistent order
3. Distribute the change list via an API
4. Provide a bootstrapping API for clients that aren’t caught up

Solution


Tracking Service
Status Updates
Changes
SQL
Postgres 9.x
Changes
protobuf
Change Server
Changes
HTTPS / JSON
Client
SQLite
"Snapshot" Server
SQL
HTTPS / JSON

Conclusion

We manage our configuration data using an RDBMS

We share it with thousands of clients easily

Limited write performance and write availability

So we limit what we use it for

Scalable read performance and high read availability

Questions?

https://github.com/apigee-labs/transicator