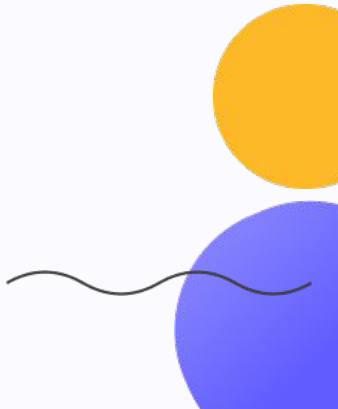
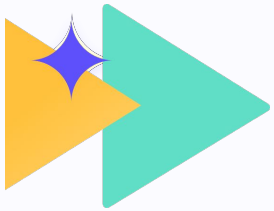
Govern Your Data Clients

The Right Way to Scale

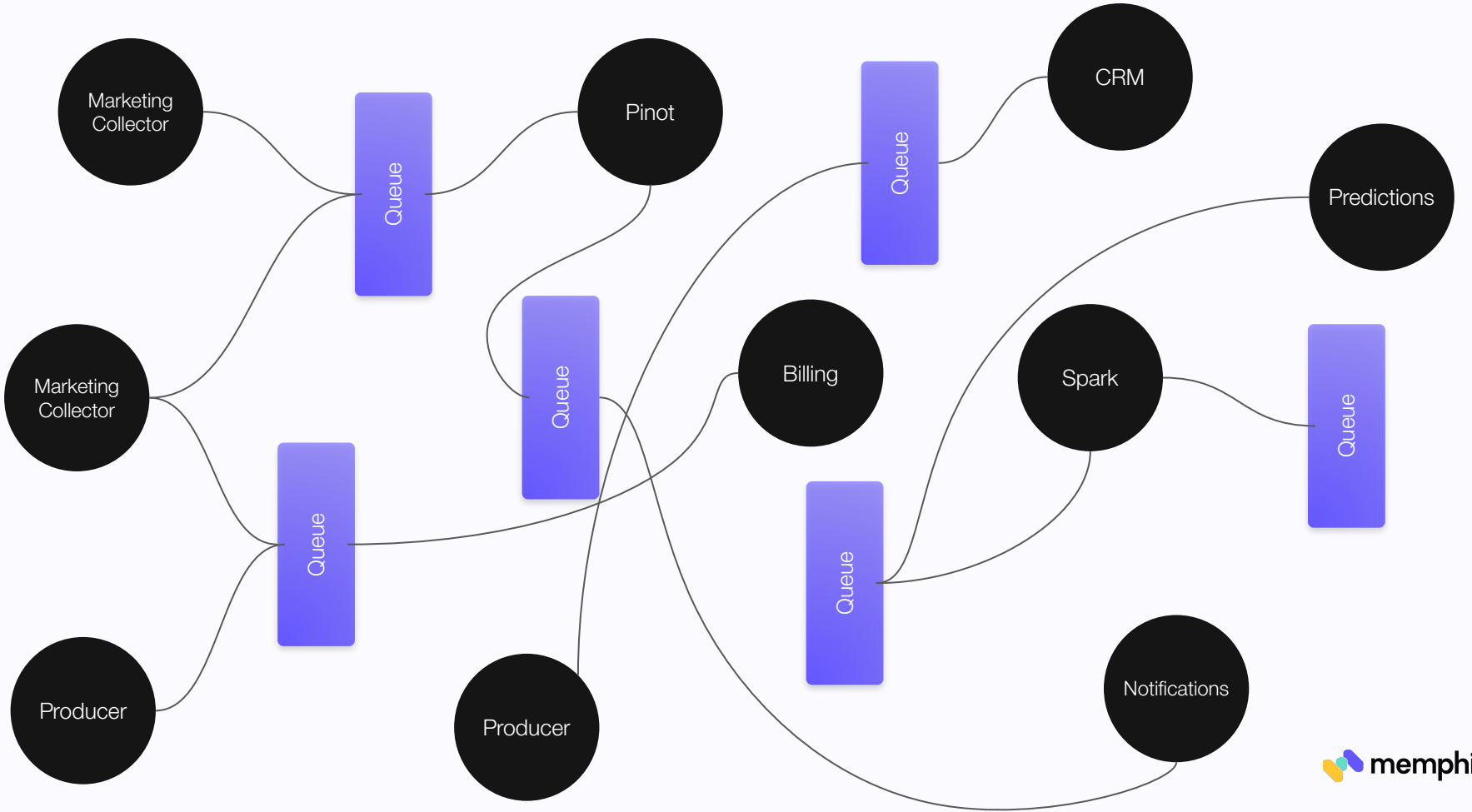


Data Council, Austin, 2023

What are data clients?

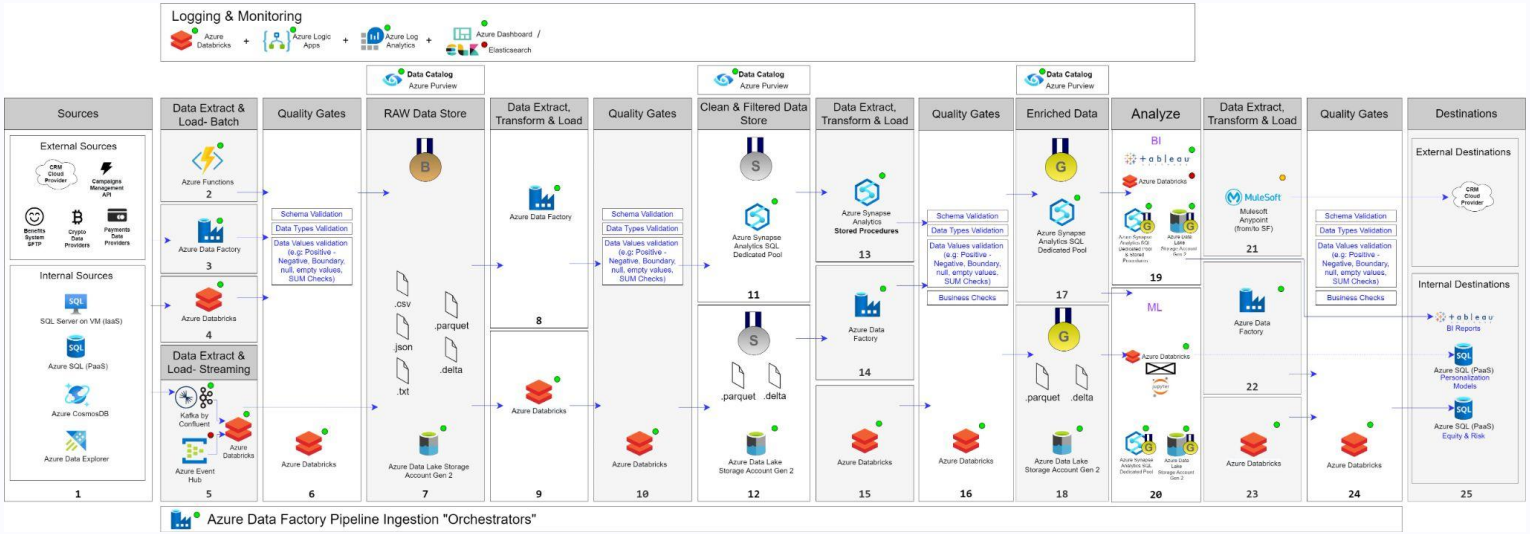


Sometimes it looks like that



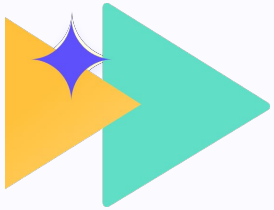
memphis

Also, like that

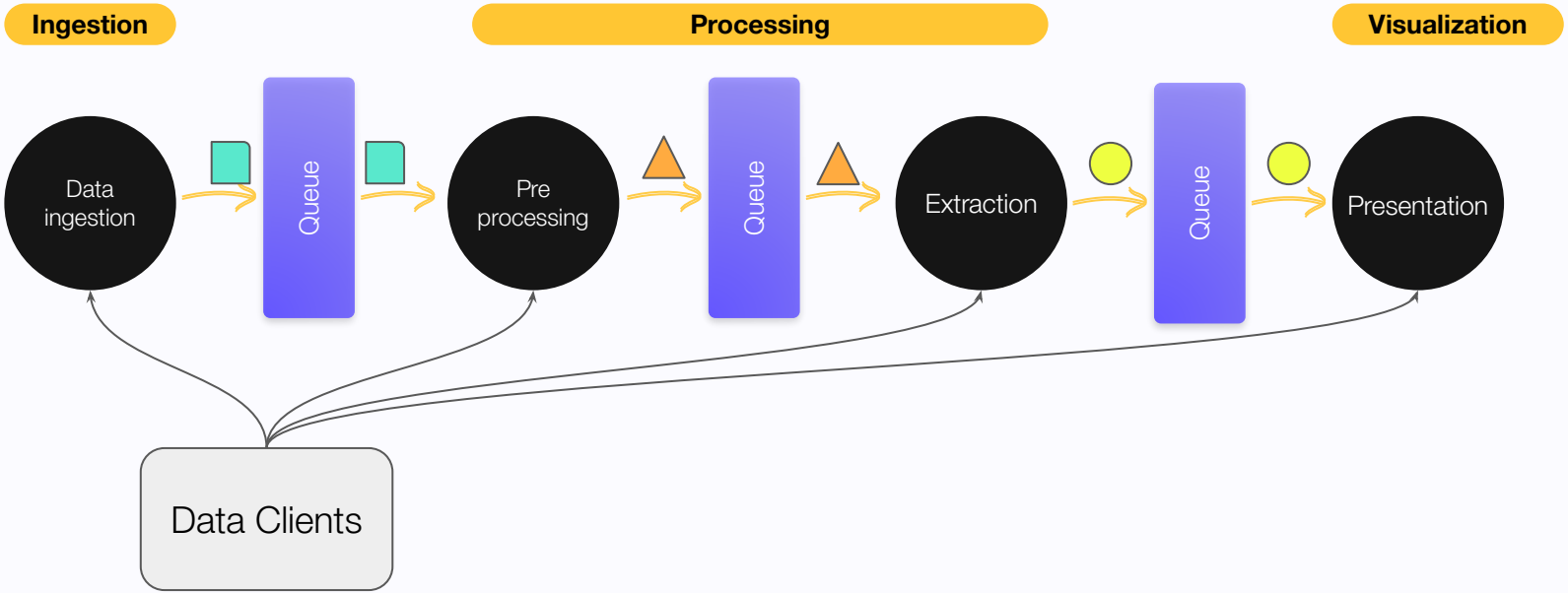


memphis

Streaming Pipelines



Streaming pipeline



memphis

As we scale,
upstream changes will happen often

309 Bytes

```
const BIG_QUERY_TABLE_FIELDS = [
    { name: 'uuid', type: 'STRING' },
    { name: 'id', type: 'STRING' },
    { name: 'event', type: 'STRING' },
    { name: 'properties', type: 'STRING' },
    { name: 'elements_chain', type: 'STRING' },
    { name: 'person', type: 'STRING' },
]
```

543 Bytes

```
const BIG_QUERY_TABLE_FIELDS = [
    { name: 'uuid', type: 'STRING' },
    { name: 'id', type: 'STRING' },
    { name: 'event', type: 'STRING' },
    { name: 'properties', type: 'STRING' },
    { name: 'elements_chain', type: 'STRING' },
    { name: 'person', type: 'STRING' },
    { name: 'elements', type: 'STRING' },
    { name: 'set', type: 'STRING' },
    { name: 'set_once', type: 'STRING' },
    { name: 'distinct_id', type: 'STRING' },
]
```

801 Bytes

```
const BIG_QUERY_TABLE_FIELDS = [
    { name: 'uuid', type: 'STRING' },
    { name: 'id', type: 'STRING' },
    { name: 'event', type: 'STRING' },
    { name: 'properties', type: 'STRING' },
    { name: 'elements_chain', type: 'STRING' },
    { name: 'person', type: 'STRING' },
    { name: 'elements', type: 'STRING' },
    { name: 'set', type: 'STRING' },
    { name: 'set_once', type: 'STRING' },
    { name: 'distinct_id', type: 'STRING' },
    { name: 'distinct__ids', type: 'STRING' },
    { name: 'team_id', type: 'INT64' },
    { name: 'ip', type: 'STRING' },
    { name: 'site_url', type: 'STRING' },
    { name: 'timestamp', type: 'TIMESTAMP' },
    { name: 'type', type: 'STRING' },
    { name: 'is__identified', type: 'STRING' },
    { name: 'bq_ingested_timestamp', type: 'TIMESTAMP' },
]
```

memphis

In good cases, you get

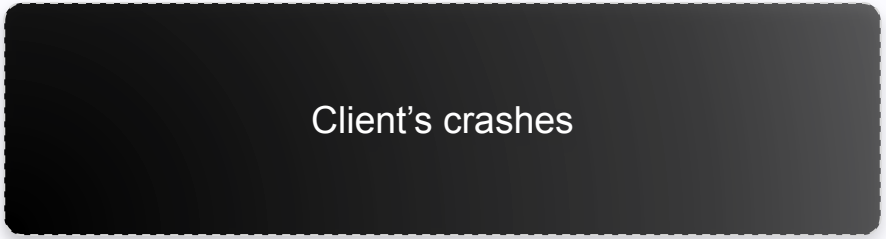
XXX% of waste traffic

Drops, extra hops, if statements, higher latency

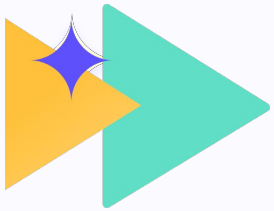
Higher costs

memphis

In bad cases, you get

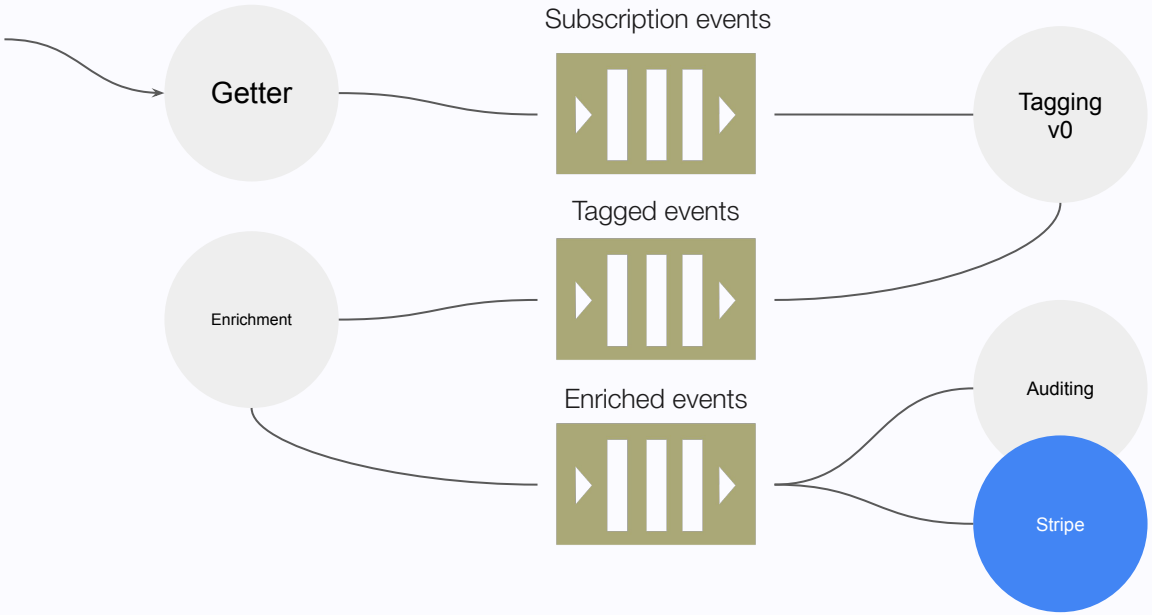


An unexpected journey of event



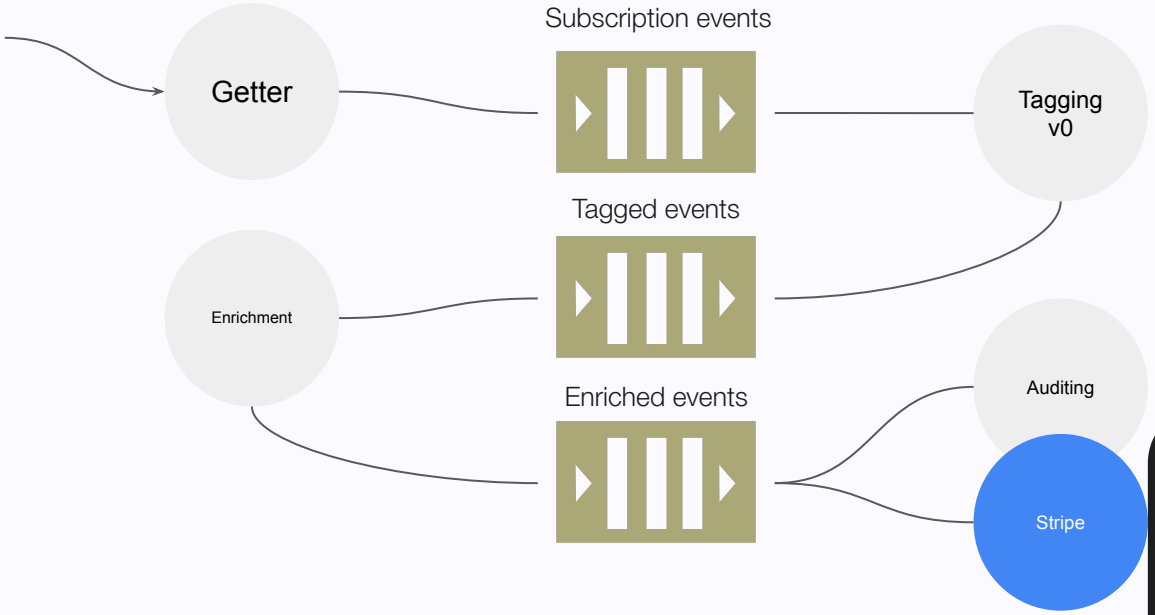
“Infrastructure looks ok, but stripe is not consuming data / messages get redelivered,
Please figure it out”

Data-level issue (Upstream changes)



memphis

Data-level issue (Upstream changes)

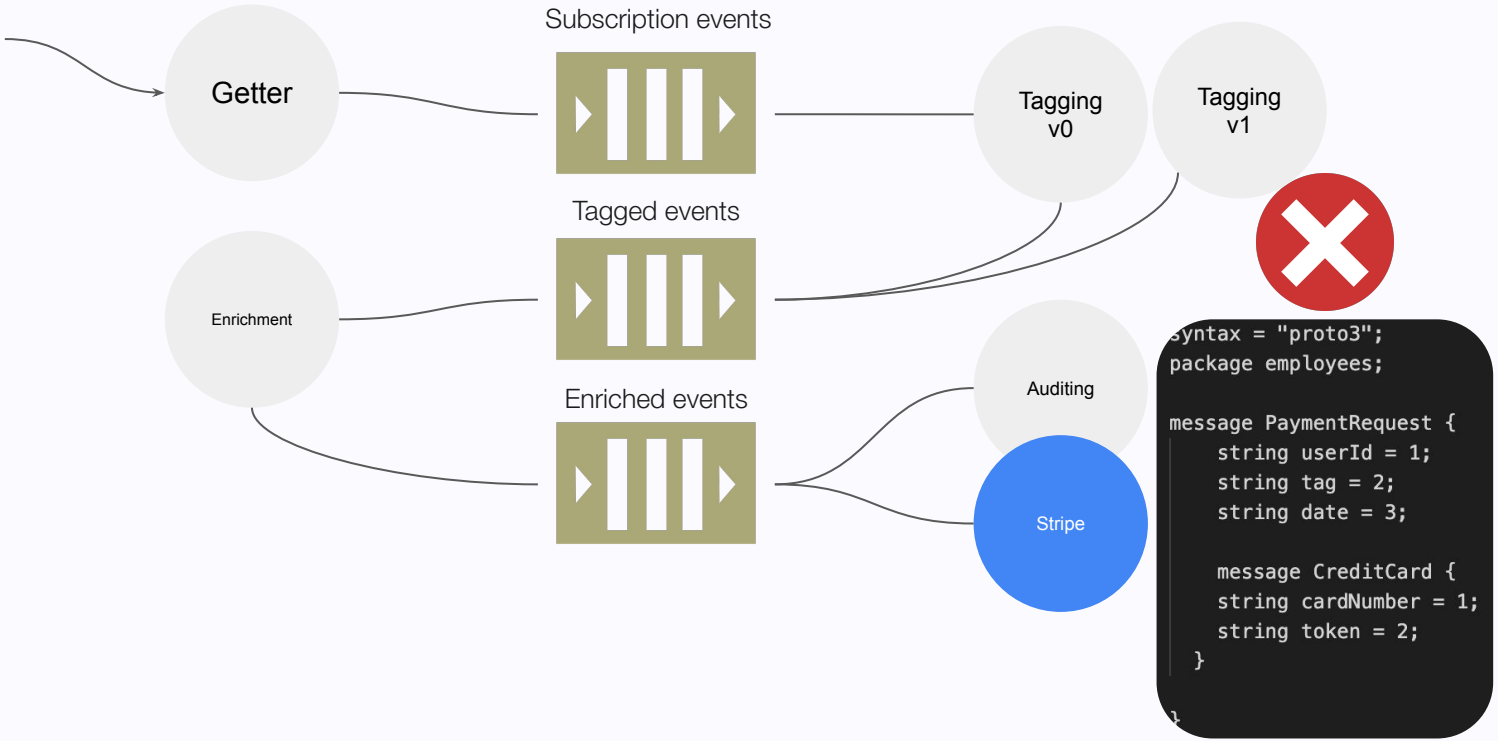


```
syntax = "proto3";
package employees;

message PaymentRequest {
    string userId = 1;
    string tag = 2;
    string date = 3;
    string cardNumber = 4;
}
```

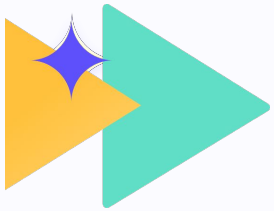
memphis

Data-level issue (Upstream changes)



memphis

How to avoid?

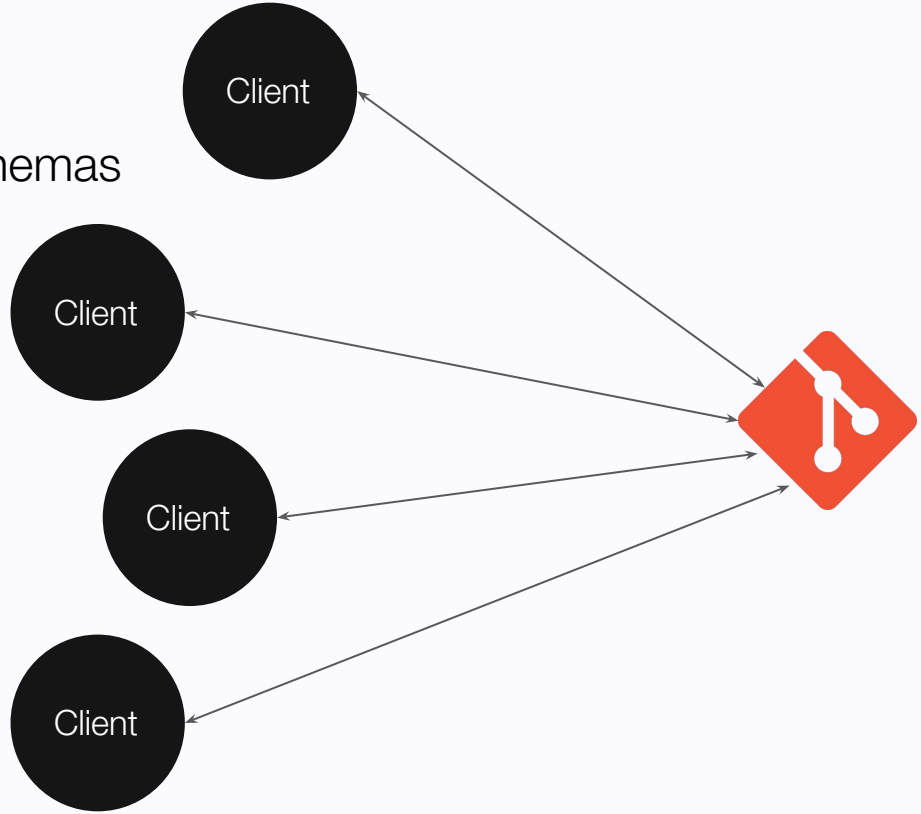


3 simple steps

- Establish a central place for your schemas.
- Create a schema.
- Enforce it.

How

- Create a git monorepo as your schemas store.
- Create a “broadcast” server that Will publish the schemas to the different clients using ws sockets.
- Make sure each client check for updates every couple of seconds.
- Use protobuf/Avro and implement serialization functions.
- Educate your team members!



memphis

Govern Your Data Clients

Thank you!

Yanivbh1 ybenhemo yaniv@memphis.dev