MessageBird

dealing with personal data

I promise it's not a GDPR talk

Messagebird

“Communication, solved”, trying to make thing easier when communicating through Voice, SMS, Chat & more.

Working in the data team.
Mostly about data processing at - *our* - scale and provide advance insights on many topics.

MessageBird it’s also

+10 other teams working on widely different products, with:

- Their own storage, at some point there is no more “one fit all”
- Many internal services in Go
- Dealing with a lot of personal data

Data team is about insight and cold storage but we can’t centralize everything.

So what about that talk?

How to handle personal data in the respect of EU laws?
How to do it with minimal engineering effort?

All that in Go *of course*.

Oh boï

Boiling done to two main problems:

Showing **precisely** what kind of personal data is transiting across all services.

Allowing the “**exercise of rights**” for every customers against theirs personal data.


A FEW
MOMENTS LATER

~~Ab~~using the existing

We already heavily use **Prometheus** and sorted out auto-discovery for it.

You can publish any kind of metrics, it’s only exposing plain text over HTTP.

You can query Prometheus data easily

First let’s map relationship

Engineers are already used to declare monitoring metrics

Now they also declare metrics about what there are **transmitting** to other services.

In Prometheus world it looks like:

```
transmit{
    "team"="data",
    "service"="sentinel",
    "to_service"="hammer",
    "data_type"="ip"
}
```

Then storage

As we declare to which service we transmit data we declare where we store it

In prometheus world it looks like:

```
store{
    "team"="data",
    "service"="bigtable",
    "data_type"="ip"
}
```

But why not etcd or <your favorite consensus service>?

We know how to operate Prometheus well:

- high availability
- cross-region
- closely monitored
- not directly data team responsibility (the best part)!
- Time, we want to achieve everything in a week or so

Tada

We can answer **where**, **when** and by **whom** personal data are processed.

At that point we built a central service, in Go, to generate a graph we can query.

What does it look like for humans?

Requests | Data Map

Reset visualisation

Filter by team:

Select...

Filter by service:

Select...

Filter by data type:

Select...

SELECTION

Select a service or link to display detailed info here.

LEGEND

- Black Widow
- Enchantress
- Gunslinger
- Jessica Drew
- Jimmy Woo
- Meltdown
- Omega Sentinel
- Rockslide
- Stilt-Man

Requests | Data Map

Reset visualisation

Filter by team:

Jessica Drew

Filter by service:

Select...

Filter by data type:

Select...

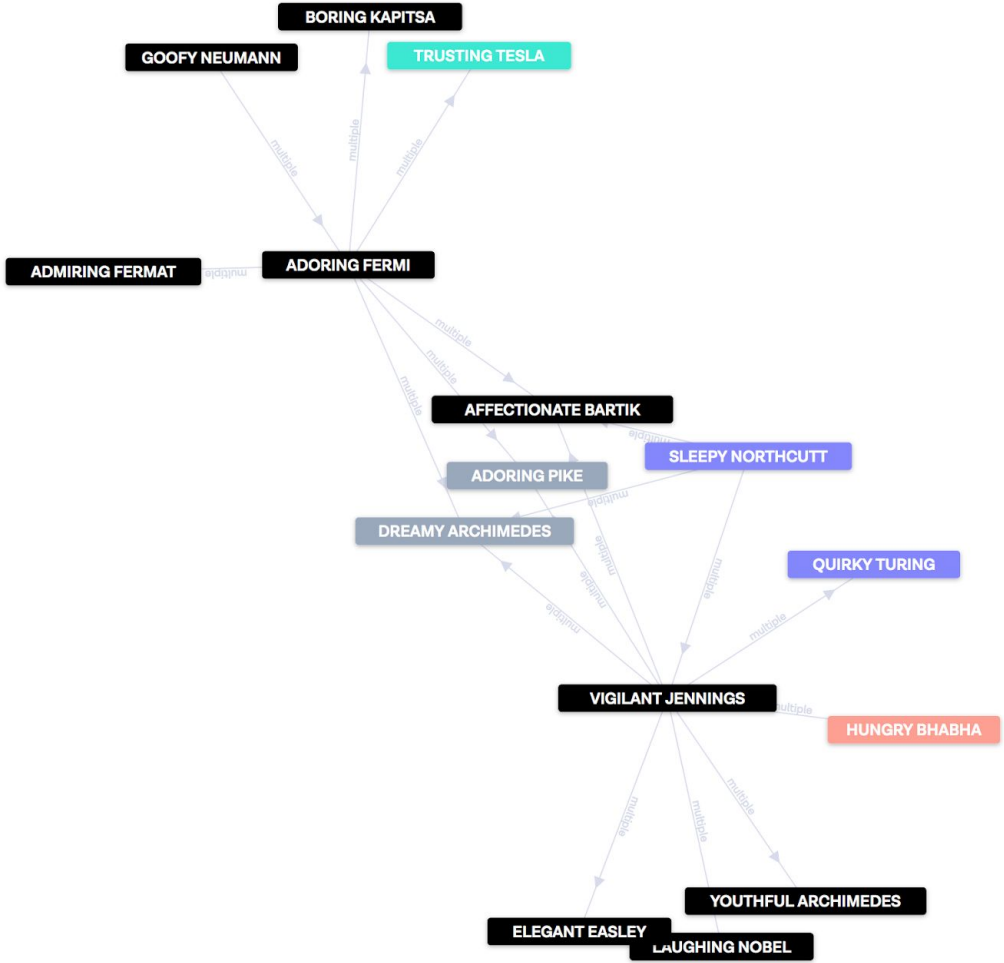
SELECTION

From team:
Sleepy Northcutt

To Team::
Dreamy Archimedes

DATA TRANSMITTED:

ip
ip
name
ip
ip
address
address
ip
ip
name



Delegating the trust for the “exercise of rights”

Widely different stacks mean one team can’t know everything.
Teams can declare “**client**” the same way they publish metadata about personal data.
Those clients are reached by a central server when needed.

In prometheus world it looks like:

```
client{
    "team"="data",
    "address"="pd-client.data.svc:1337",
    "protocol"="grpc"
}
```

Contracts, contracts

When a team declares a “client” they must implement a well defined API. We use protobuf & gRPC to establish clear contract

```
service PDClient {
    rpc Delete(DeleteRequest) returns (DeleteResponse);
    rpc Get(GetRequest) returns (GetResponse);
}
```

And automatically generate an HTTP API from it with grpc-gateway

Let’s sum-up

We a have:

- A queryable graph of where, when and by whom the data is moving, is processes and stored
- A list of clients implementing a well defined API

Each time a customer uses their rights of getting, deleting personal data we can automatically pinpoint which team needs to take actions.

Giving some slack

We created a placeholder “client” to dispatch incoming requests to our tickets system and notify only concerned teams.

Teams can automate the process on their own time.

Was it worth it?

Designing, implementing and mapping most of personal data took only a bit more than a week.

The automation allowed to rollout a public API to deal with deleting and getting customer personal data.

MessageBird

Questions

felix@messagebird.com
(we are always hiring)