Hook-In Privacy Capabilities for gRPC

Jiaao Li, Riccardo Marin, Timo Ramsdorf

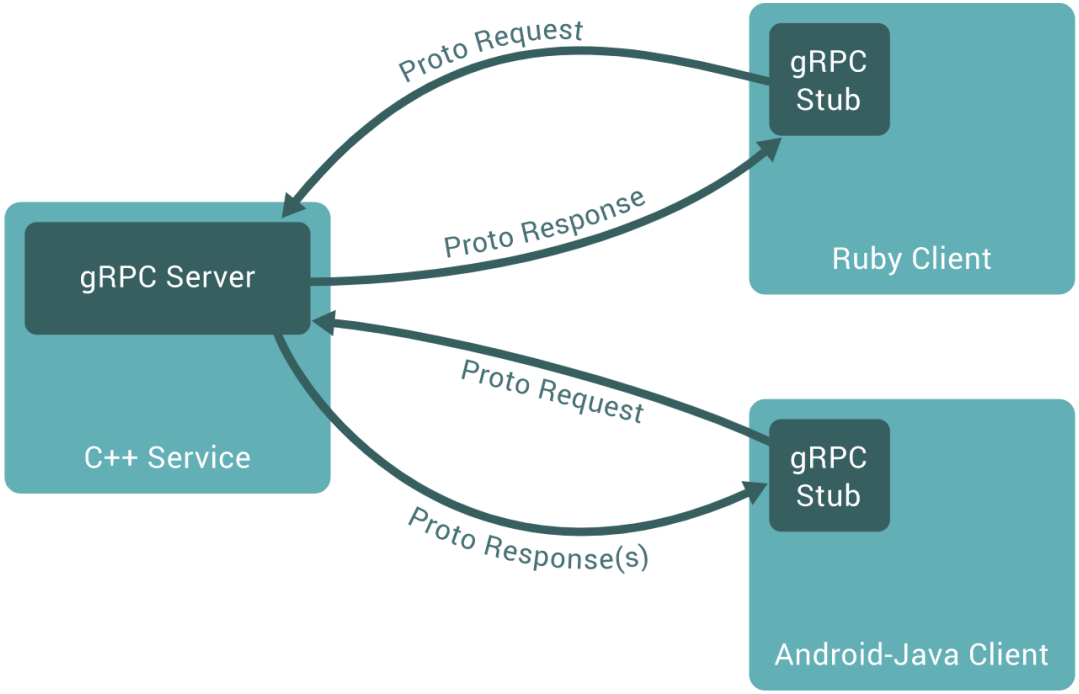
Privacy Engineering, TU Berlin

31st May 2022

Overview

Background

gRPC: connecting microservices with a high performance Remote Call Procedure framework



↓

Any privacy tools for data subjects and data controllers?

State of the art

Related work (not necessarily gRPC-based)

- *"Towards Application-Layer Purpose-Based Access Control"*[1],
  - PBAC at application level (ORM)
  - Developer-friendly purposes configuration (YAML)
  - Acceptable performance overhead
- *"Per-Query Data Minimization for Privacy-Compliant Web APIs"* [2]
  - Attribute based access control for GraphQL API
  - Information reduction techniques
  - Reasonable performance results

[1] 2020, F. Pallas, M. Ulbricht, S. Tai, T. Peikert, M. Reppenhagen, D. Wenzel, P. Wille, K. Wolf

[2] 2022, F. Pallas, D. Hartmann, P. Heinrich, J. Kipke, E. Grünewald

State of the art

Related work (gRPC-based)

- *"Implementing Data Flow Assertions in gRPC and Protobufs – Final Report"* [3]
  - Library in Go with encryption and policy checks
  - Significant impact on performance
- *"Framework for Data Tracking across Data Controllers and Processors"* [4]
  - Graphs for tracking data flow
  - Missing performance evaluation

[3] 2020, A. Mahajan, Y. Xue, J. Weisskoff

[4] 2020, Z. Lai, Y. Xin, A. Yu

State of the art

- more...

We couldn't find a **comprehensive, re-usable gRPC framework** offering **privacy capabilities**, such as data minimization and access control

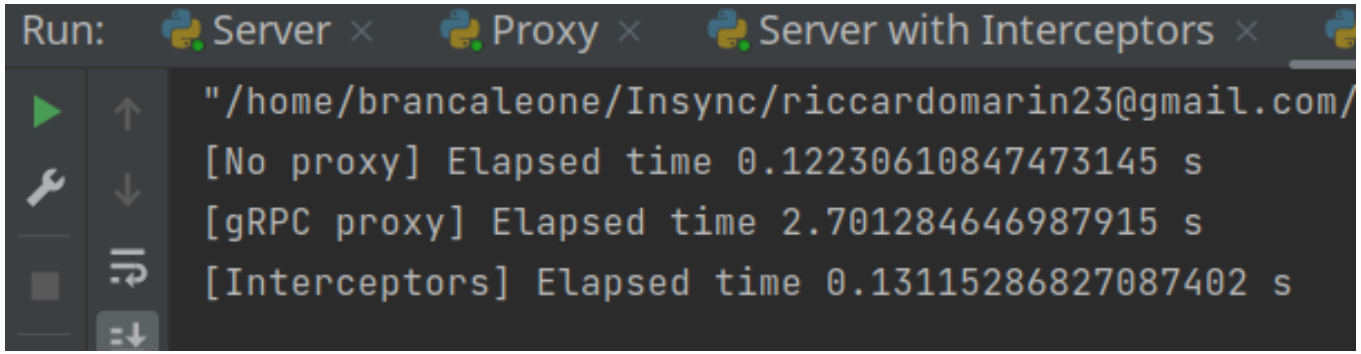
The component(s)

We are going to implement and benchmark 3 different technical solutions:

1. Interceptors
2. Classical proxy
3. Binary stream proxy

The component(s)

Preliminary results



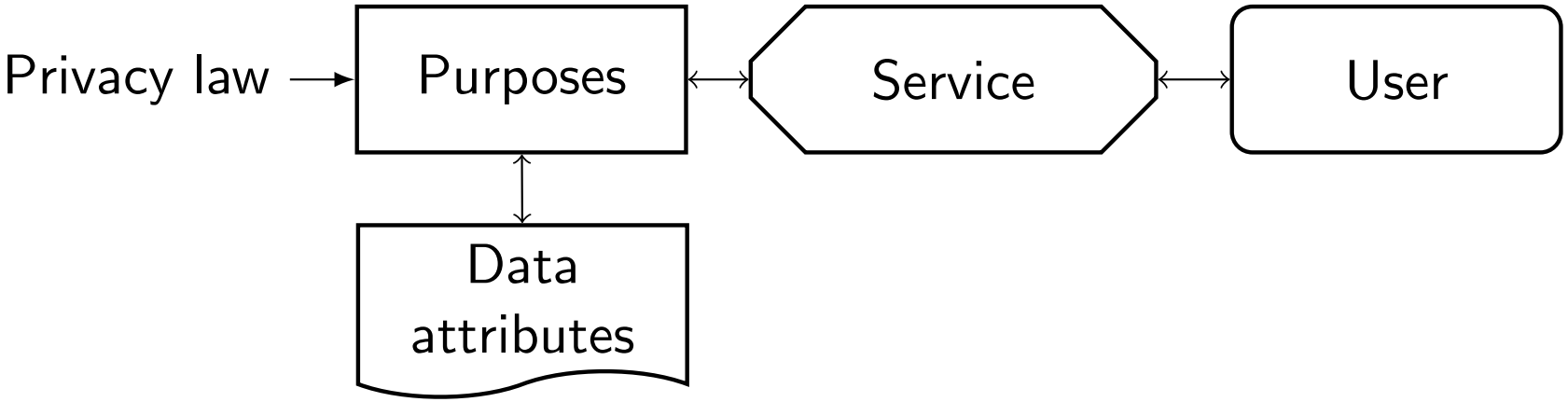
Implementation

We expect to implement the following **data minimization** techniques:

- Erasure of data field
- Generalization
- Noising
- Hashing

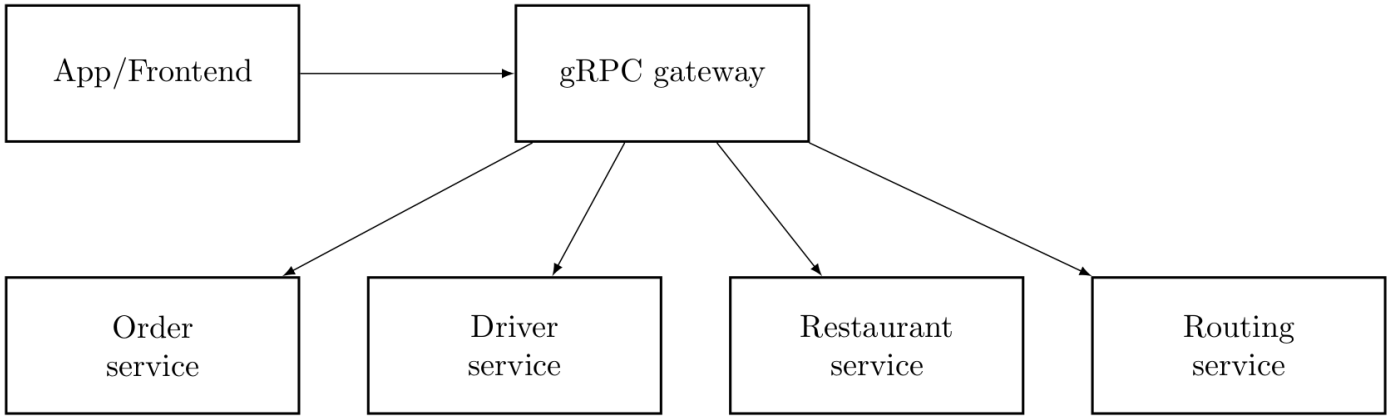
Implementation

In addition, an **access control scheme** will be supported:
Purpose-based Access Control (PBAC)



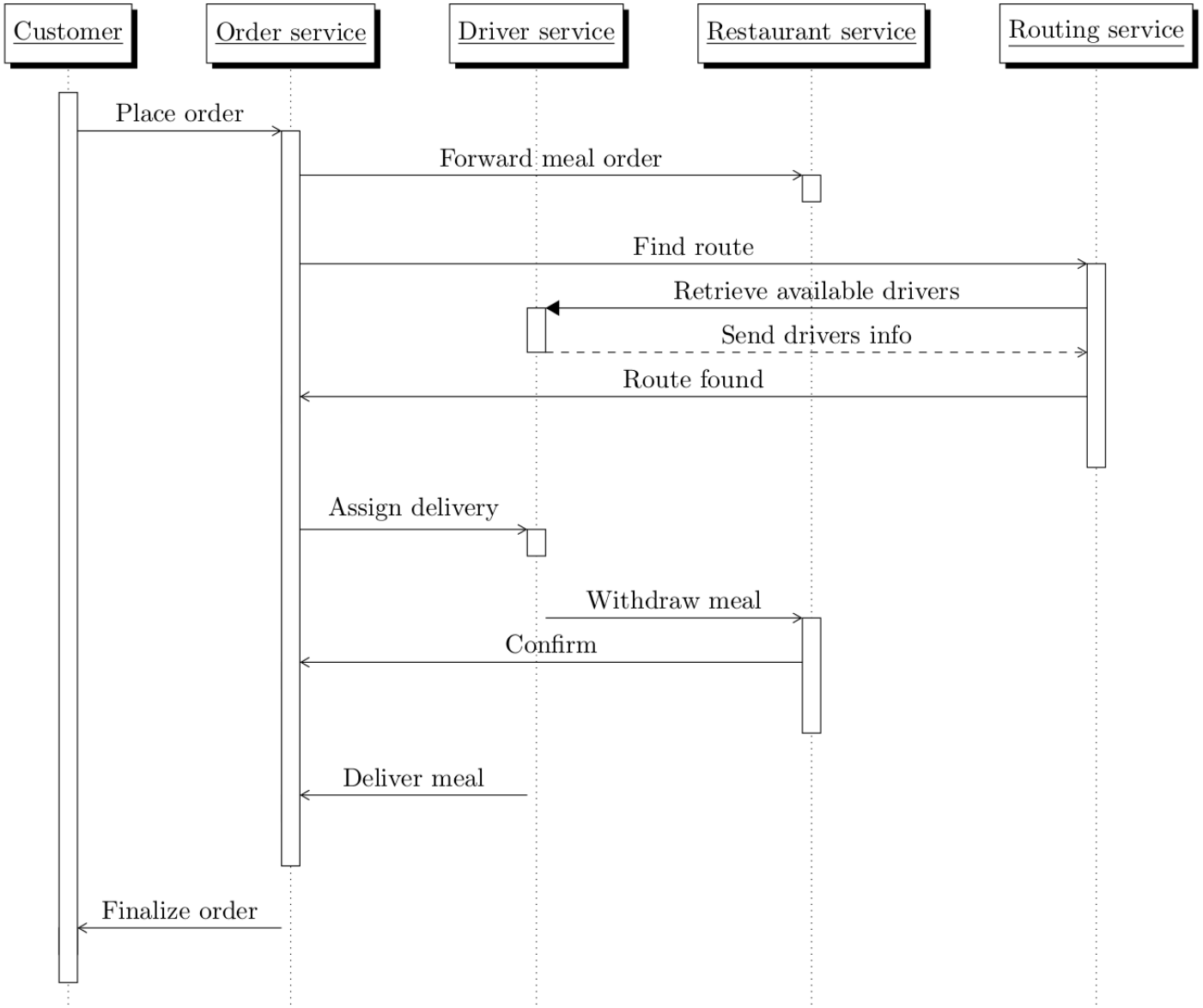
Application

The proof of concept will be tested in a **food delivery app** context



Application

Model of the communication between services



Application

Data accessed by parties and related data minimization measures

Service	**Personal data**	**Privacy tools**
Order service	Name	
	Surname	
	Favorite addresses	
	Current position	
	Orders history	
Driver service	Meal	Data erasure (he doesn't know about the meal)
	Customer data (except delivery address)	Data erasure (no info about customer except for address)
	Delivery address	
Restaurant service	Customer data (except delivery address)	Data erasure (no info about customer except for address)
	Delivery address	Generalization (convert to distance)
	Meal	
	Driver identity (name, . . .)	Hashing (restaurant can see only a pseudonymized ID)
Routing service	Driver position	Noising (return an approximate position)

Application

Purposes and related data and services

$$\text{Order service} \rightarrow \{\textit{Meal purchase}\}$$
$$\text{Driver service} \rightarrow \{\textit{Meal delivery}\}$$
$$\text{Restaurant service} \rightarrow \{\textit{Meal cooking}, \textit{Meal collection}\}$$
$$\text{Routing service} \rightarrow \{\textit{Route computation}\}$$

Purpose	**Personal data**
Meal purchase	Name
	Surname
	Delivery address
Meal delivery	Delivery address
Meal cooking	Meal
	Distance
Meal collection	Driver id
Route computation	Driver position

Q&A

Q&A Time