salesforce

Salesforce RDF extension

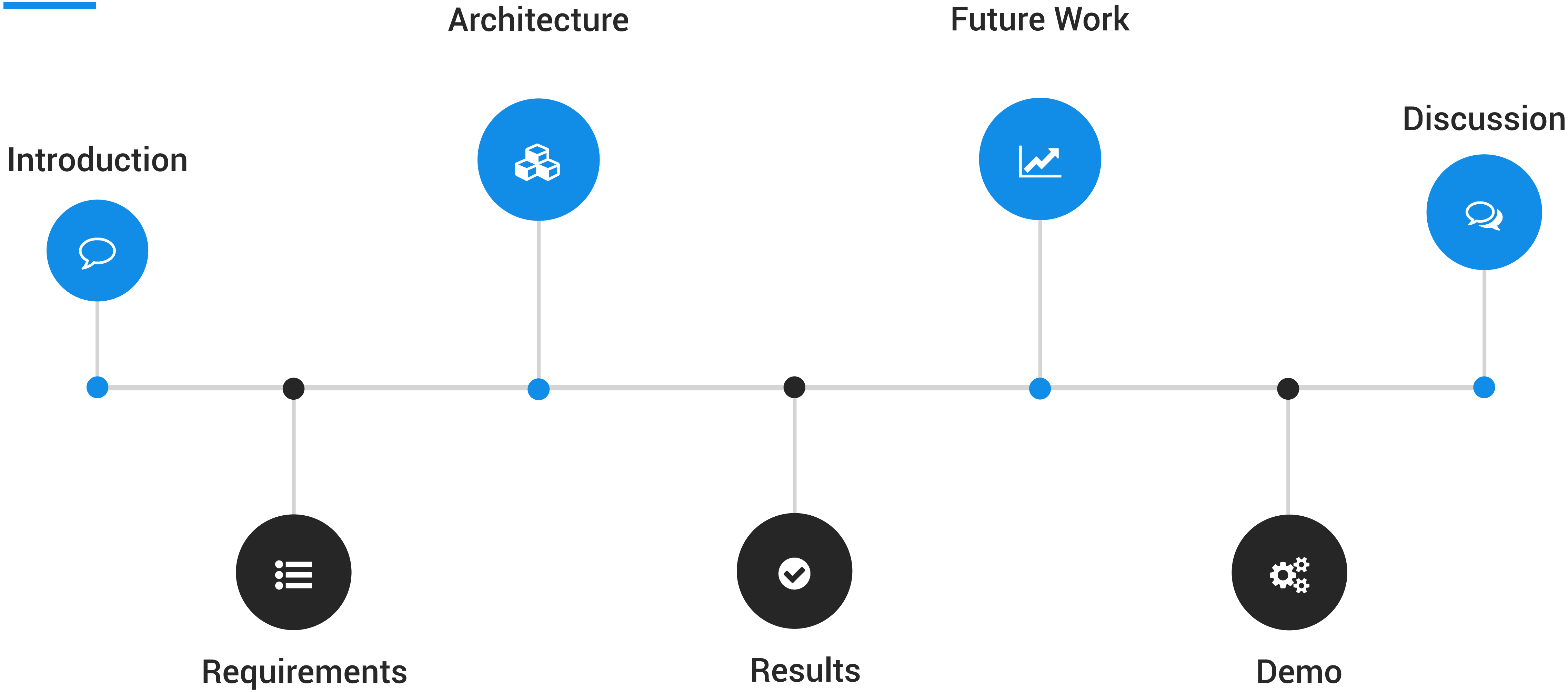
Niklas Petersen

Omar Rana

Mintcho Tzatzarov

Alexander Melnyk

Outline



Introduction

Requirements

Architecture

Results

Future Work

Demo

Discussion

Introduction

Introduction

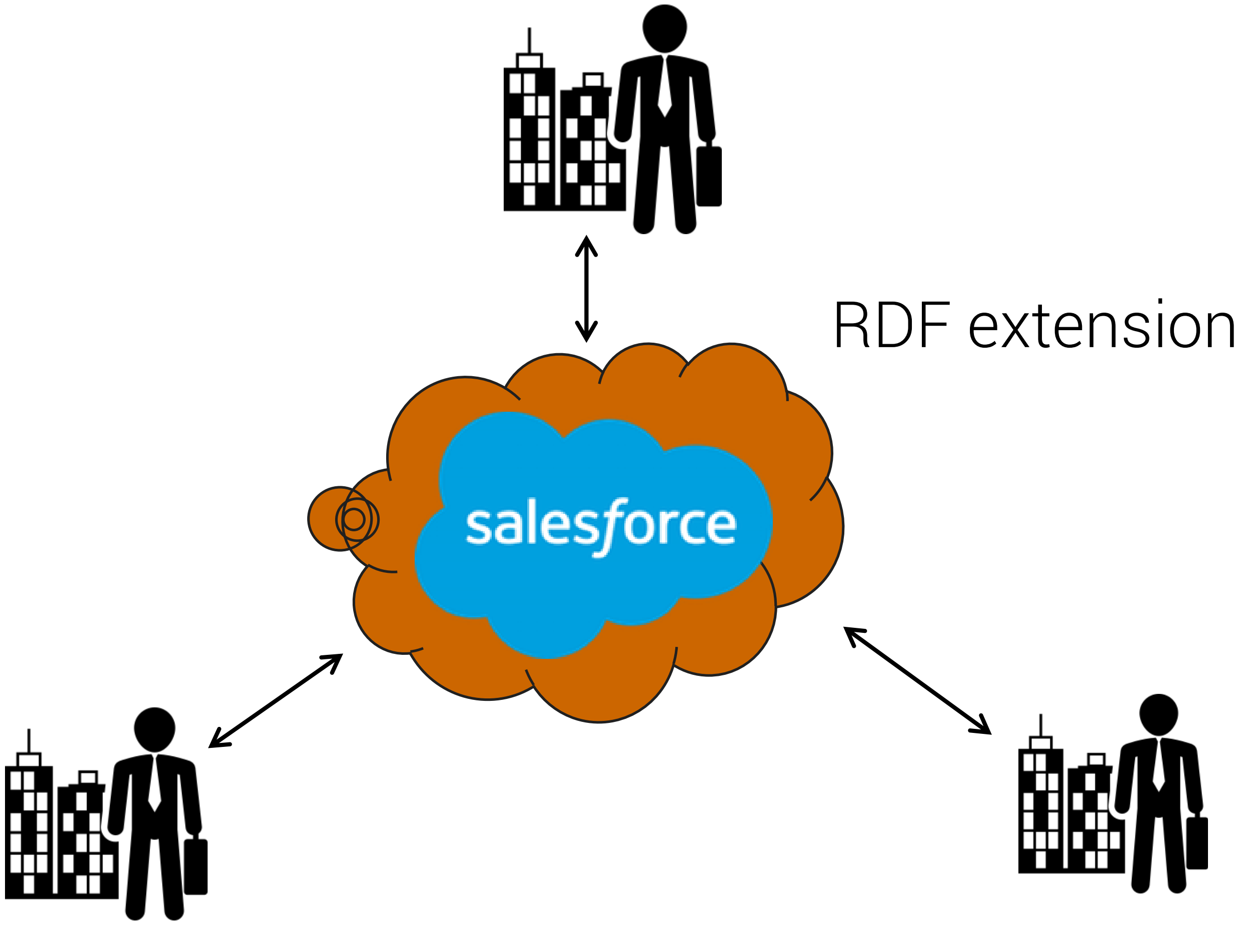
LUCID Project
Linked valUe ChaIn Data

Development of distributed networks of supply chain partners in order to exchange data about their work, their organization and their products.

Salesforce

Cloud computing company that provides a customer relationship management platform (SaaS and PaaS).

Perspective



RDF extension

Introduction

Goal

Create a prototype to read, write and transform data from salesforce.

Requirements

Requirements

Functional

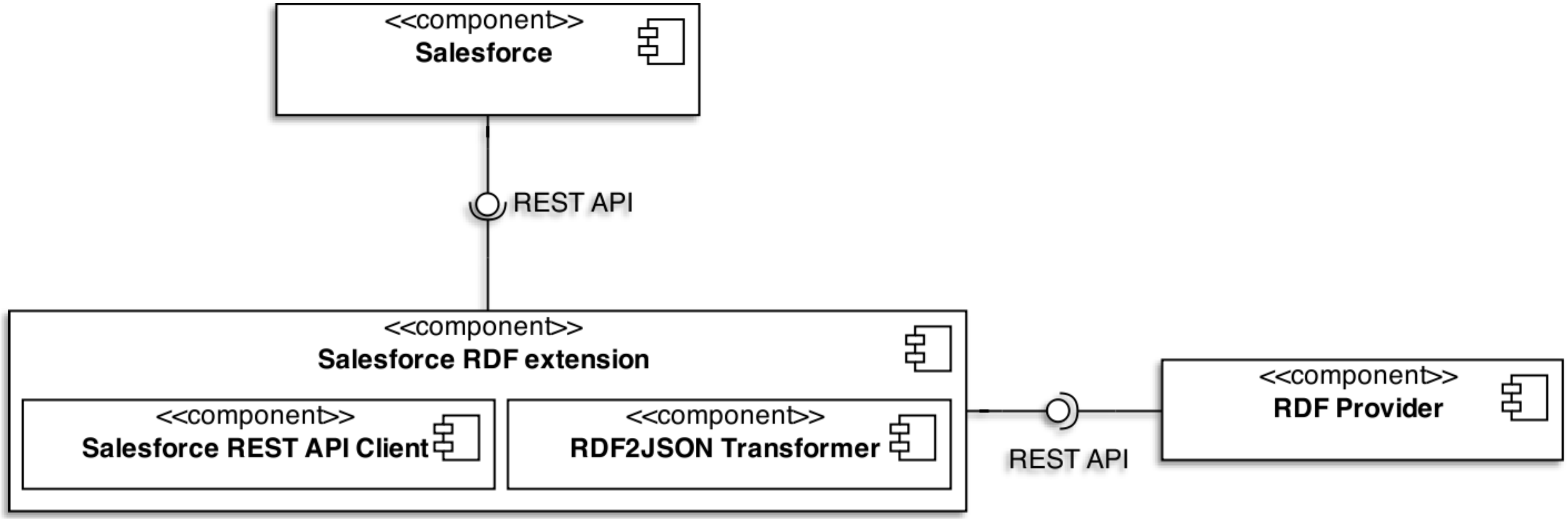
- Import
- Export
- Transformation

Nonfunctional

- Interoperability
- Usability
- Performance

Architecture

Architecture overview



Components

Salesforce REST API client

- Core communication with API
- Provide limited access

RDF2JSON transformer

- Transformation of RDF, JSON
- Based on provided input

Salesforce RDF extension

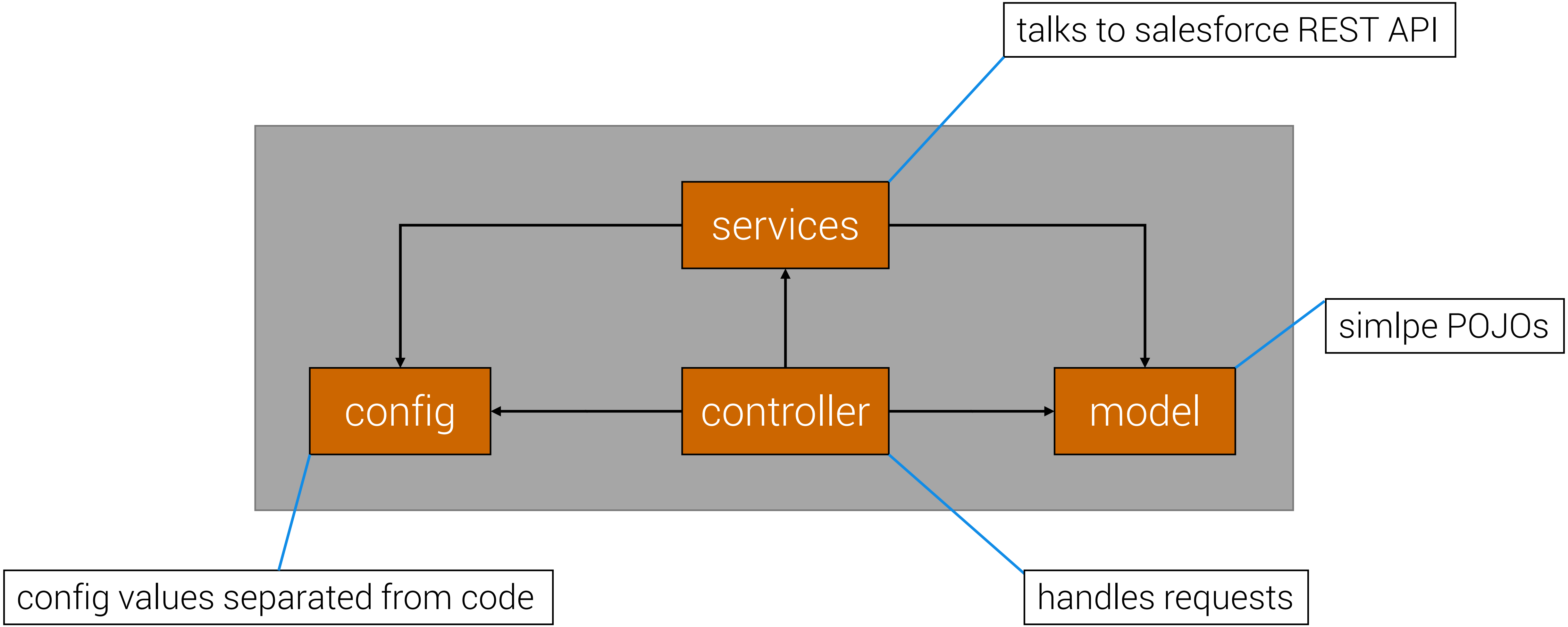
- REST Web Service
- Integration of components

RDF Provider / Test client

- Input for the extension
- Testing purposes

Results

Project Structure



services

config

controller

model

talks to salesforce REST API

simlpe POJOs

config values separated from code

handles requests

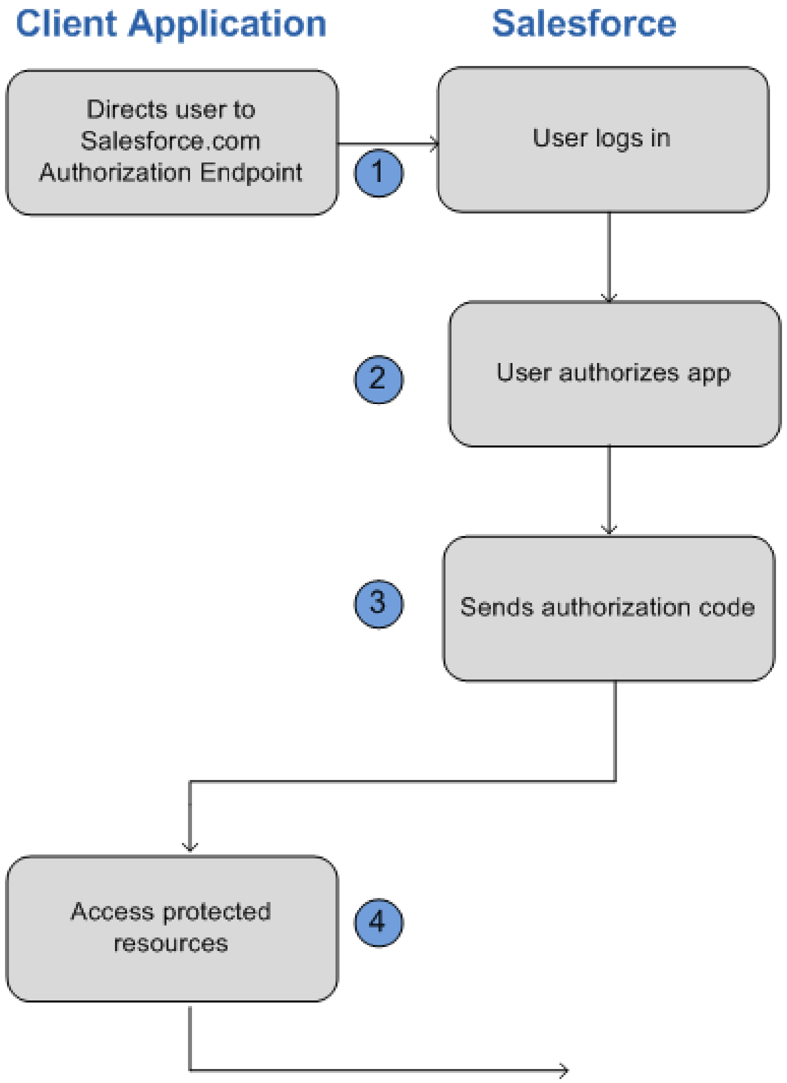
REST API Client

LoginService

User-agent authentication flow. Must be configured in salesforce as external app.

ContactService

Access Contact objects of salesforce. Implemented as simple CRUD operations.

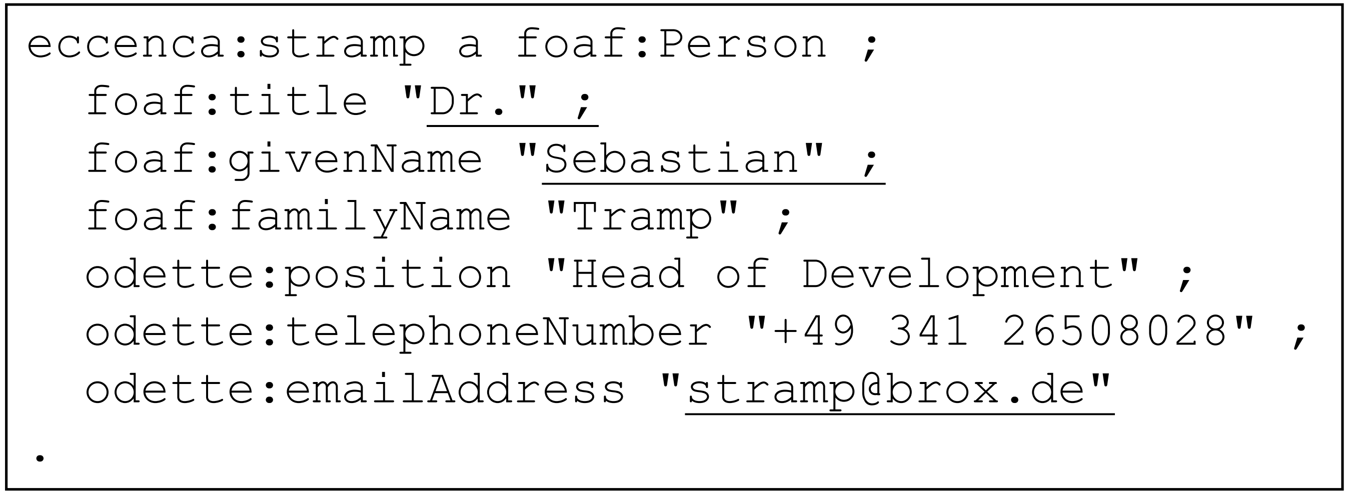


Salesforce user-agent authentication flow [1]

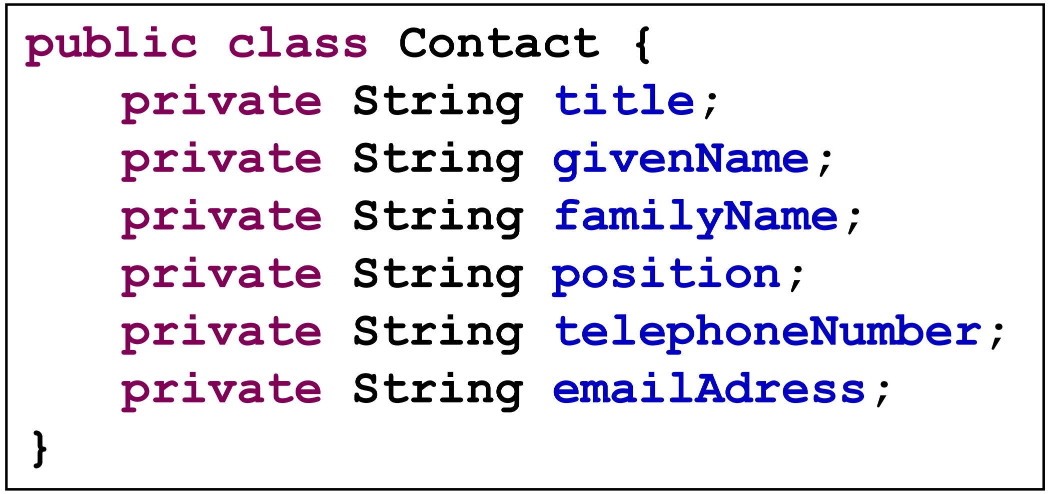
RDF Transformer

RDF → Contact

ContactService extracts **foaf:Person** types and transforms them to contact objects. Use reflection to iterate over properties.



```
eccenca:stramp a foaf:Person ;
  foaf:title "Dr." ;
  foaf:givenName "Sebastian" ;
  foaf:familyName "Tramp" ;
  odette:position "Head of Development" ;
  odette:telephoneNumber "+49 341 26508028" ;
  odette:emailAddress "stramp@brox.de"
.
```

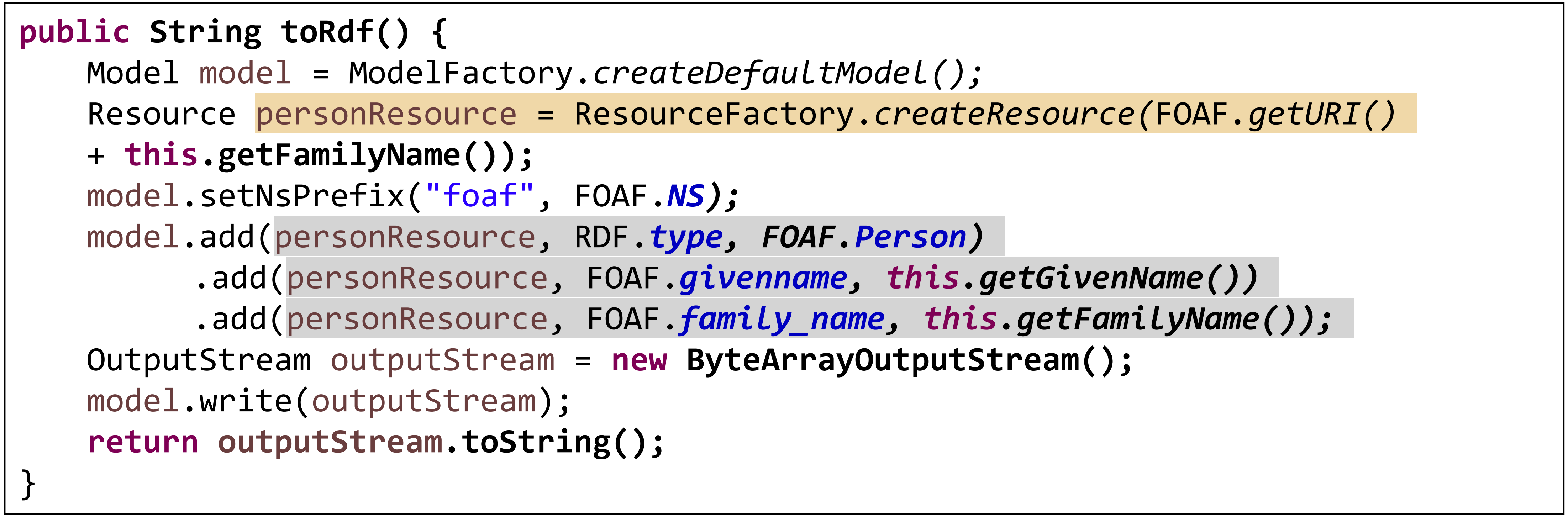


```
public class Contact {
    private String title;
    private String givenName;
    private String familyName;
    private String position;
    private String telephoneNumber;
    private String emailAdress;
}
```

RDF Transformer

Contact → RDF

Contact class implements a method to convert the instance to an RDF representation.



```
public String toRdf() {
    Model model = ModelFactory.createDefaultModel();
    Resource personResource = ResourceFactory.createResource(FOAF.getURI()
    + this.getFamilyName());
    model.setNsPrefix("foaf", FOAF.NS);
    model.add(personResource, RDF.type, FOAF.Person)
          .add(personResource, FOAF.givenname, this.getGivenName())
          .add(personResource, FOAF.family_name, this.getFamilyName());
    OutputStream outputStream = new ByteArrayOutputStream();
    model.write(outputStream);
    return outputStream.toString();
}
```

Future Work

Future work

Support more sales objects

The focus of this work was only on Contact objects. Salesforce has more objects that needs to be integrated.

Generalize RDF transformation

The implemented RDF transformer know only one vocabulary.

Provide more feedback

Limited exception handling. More feedback would make the extension robust and usable.

Demo

References

[1] Salesforce Oauth 2.0: https://developer.salesforce.com/docs/atlas.en-us.api_rest.meta/api_rest/intro_understanding_user_agent_oauth_flow.htm