json2puml

Make data visible and understandable

Jens Fudickar, April 2022

API’s and DATA

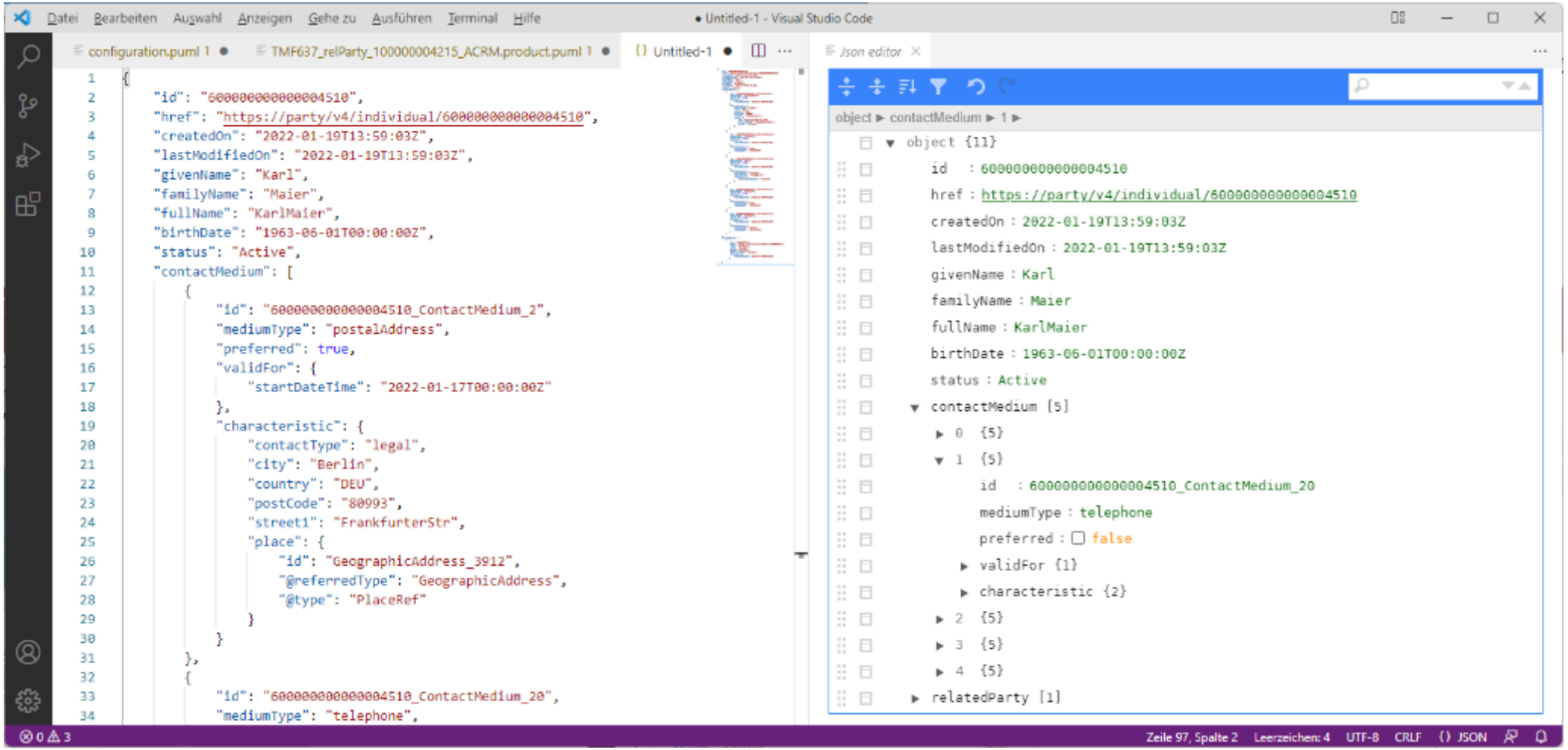
Working with API’s leads to working with JSON data

A typical answer of a TMF call can look like this:

{"id": "600000000000004510", "href": "https://party/v4/individual/600000000000004510", "createdOn": "2022-01-19T13:59:03Z", "lastModifiedOn": "2022-01-19T13:59:03Z", "givenName": "Karl", "familyName": "Maier", "fullName": "KarlMaier", "birthDate": "1963-06-01T00:00:00Z", "status": "Active", "contactMedium": [{"id": "600000000000004510_ContactMedium_2", "mediumType": "postalAddress", "preferred": true,"validFor": {"startDateTime": "2022-01-17T00:00:00Z"},"characteristic": {"contactType": "legal", "city": "Berlin", "country": "DEU", "postCode": "80993", "street1": "FrankfurterStr", "place": {"id": "GeographicAddress_3912", "@referredType": "GeographicAddress", "@type": "PlaceRef"}}},{"id": "600000000000004510_ContactMedium_20", "mediumType": "telephone", "preferred": false,"validFor": {"startDateTime": "2022-01-17T00:00:00Z"},"characteristic": {"contactType": "home", "phoneNumber": "0123457689"}},{"id": "600000000000004510_ContactMedium_60", "mediumType": "email", "preferred": true,"validFor": {"startDateTime": "2022-01-17T00:00:00Z"},"characteristic": {"contactType": "personal", "emailAddress": "jesf@gmx.de", "extensions": {"verificationStatus": "verified"}}},{"id": "600000000000004510_ContactMedium_75", "mediumType": "telephone", "preferred": true,"validFor": {"startDateTime": "2022-01-17T00:00:00Z"},"characteristic": {"contactType": "home", "phoneNumber": "0123456789"}},{"id": "600000000000004510_ContactMedium_76", "mediumType": "telephone", "preferred": true,"validFor": {"startDateTime": "2022-01-17T00:00:00Z"},"characteristic": {"contactType": "mobile", "phoneNumber": "01721234567"}}],"relatedParty": [{"id": "100000004215", "href": "https:///party/v4/relatedParty/100000004215", "name": "KarlMaier", "role": "Customer", "@type": "Customer", "@referredType": "Customer", "validFor": {"startDateTime": "2022-01-17T00:00:00Z"}}]}

How to visualize / understand this ?

(Online) JSON Editor

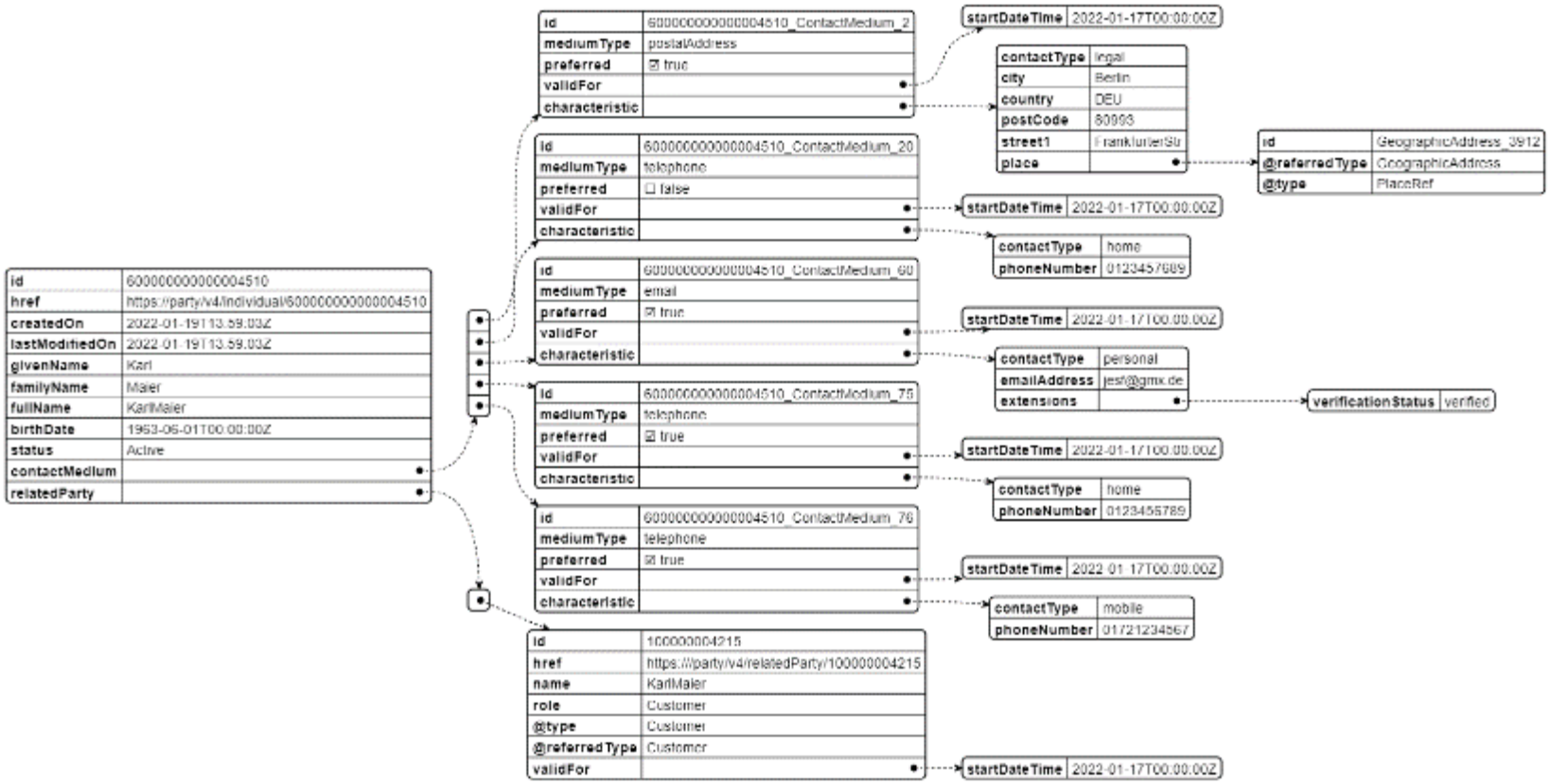


This makes the live easier but it is still complex ☺

Confluence and Standard PlantUML

PlantUML supports an OOTB visualisation of JSON.

```
@startjson
<json>
@endjson
```

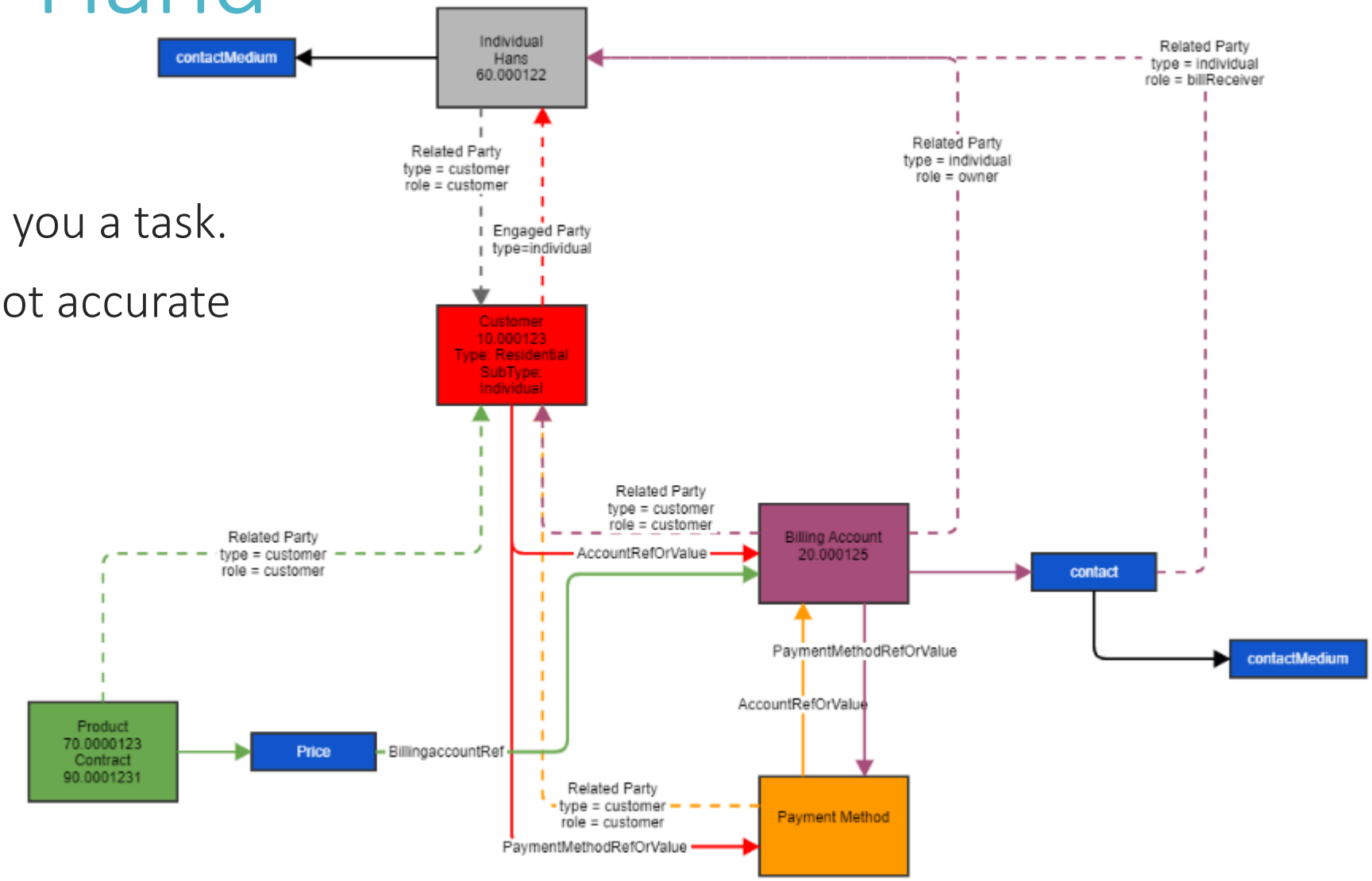


This improves also, but there is no knowledge about the data models behind

Drawing by Hand

It’s nice, and it’s giving you a task.

But it’s not effective, not accurate and not fast enough.



One Solution: `json2puml`

`json2puml` is a command line tool developed to generate PlantUML files based JSON files (TMF based).

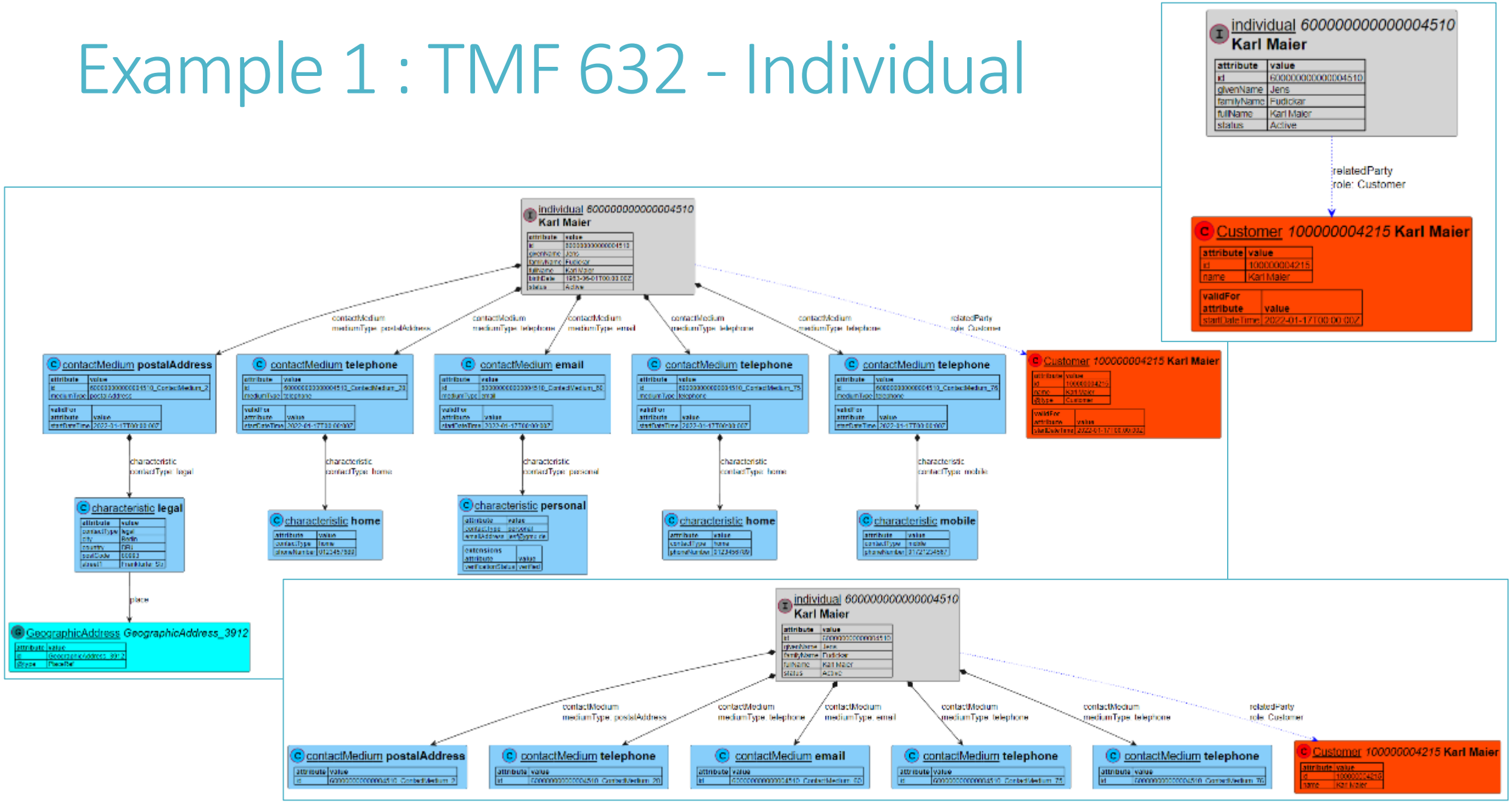
`json2puml` has an understanding of how data is structured (in TMF) and simplifies and visualises the outcome.

`json2puml` has the possibility to combine the JSON results of multiple API calls into one result set.

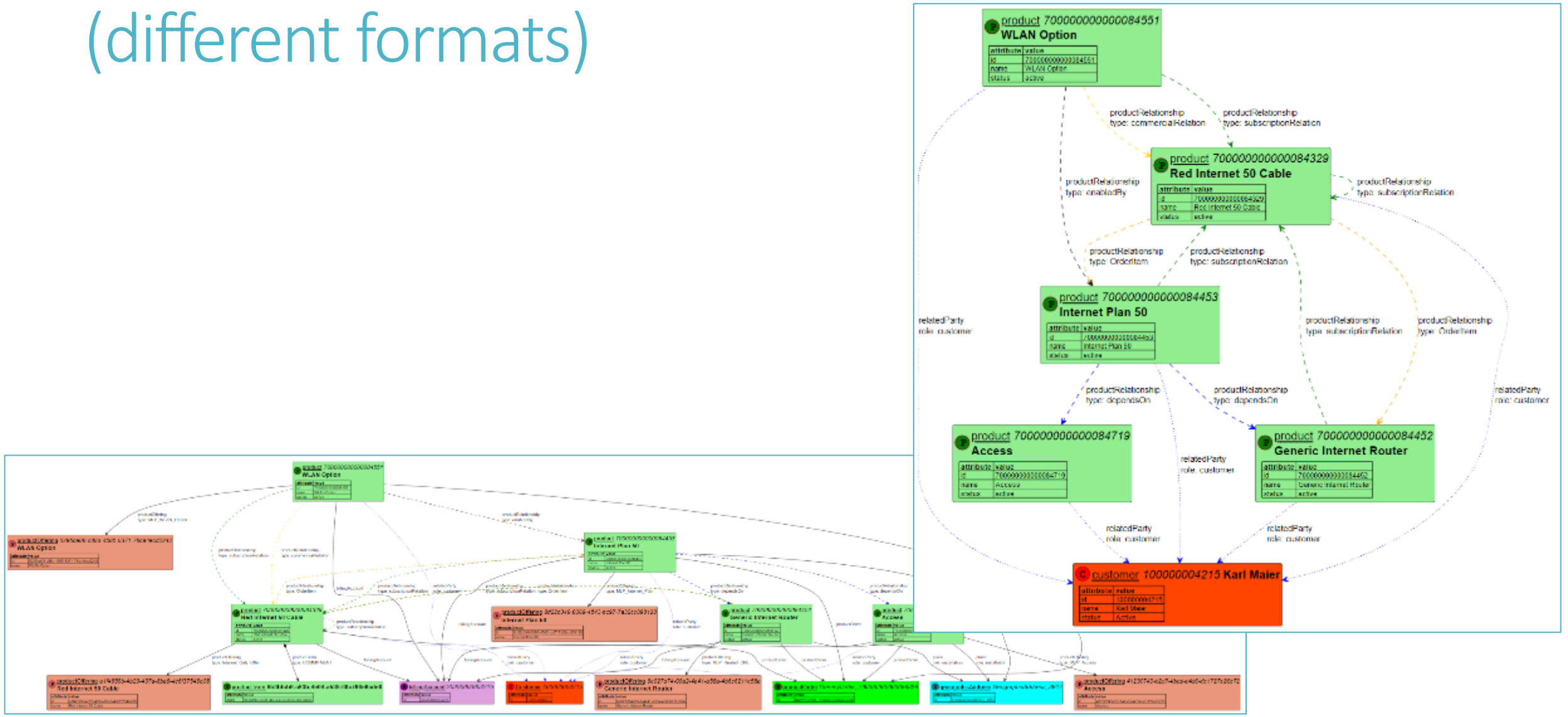
`json2puml` is highly configurable to generate outcomes in different detailed levels.

`json2puml` is free to use for everyone.

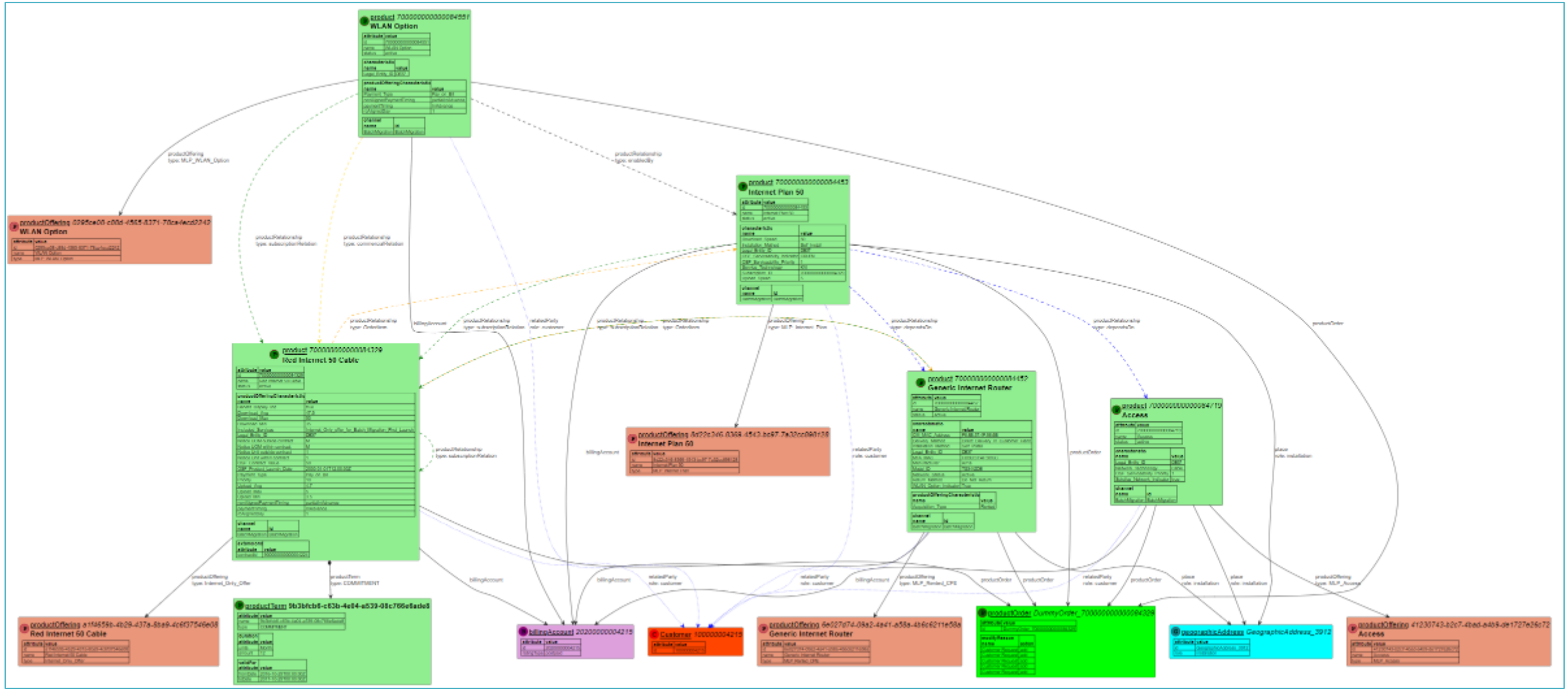
Example 1 : TMF 632 - Individual



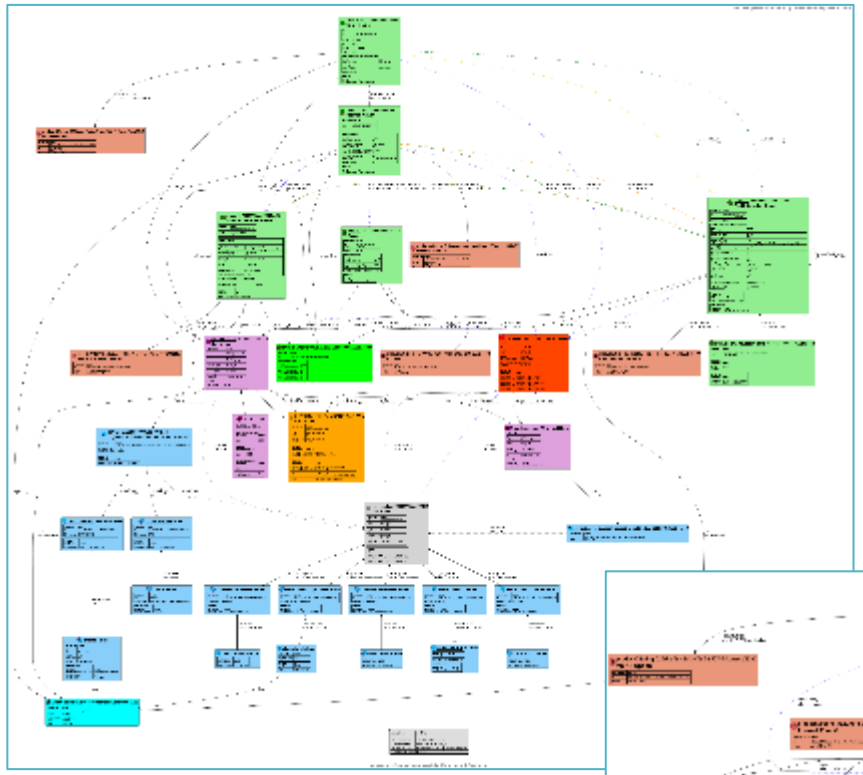
Example 2: TMF 637 – All products of a customer (different formats)

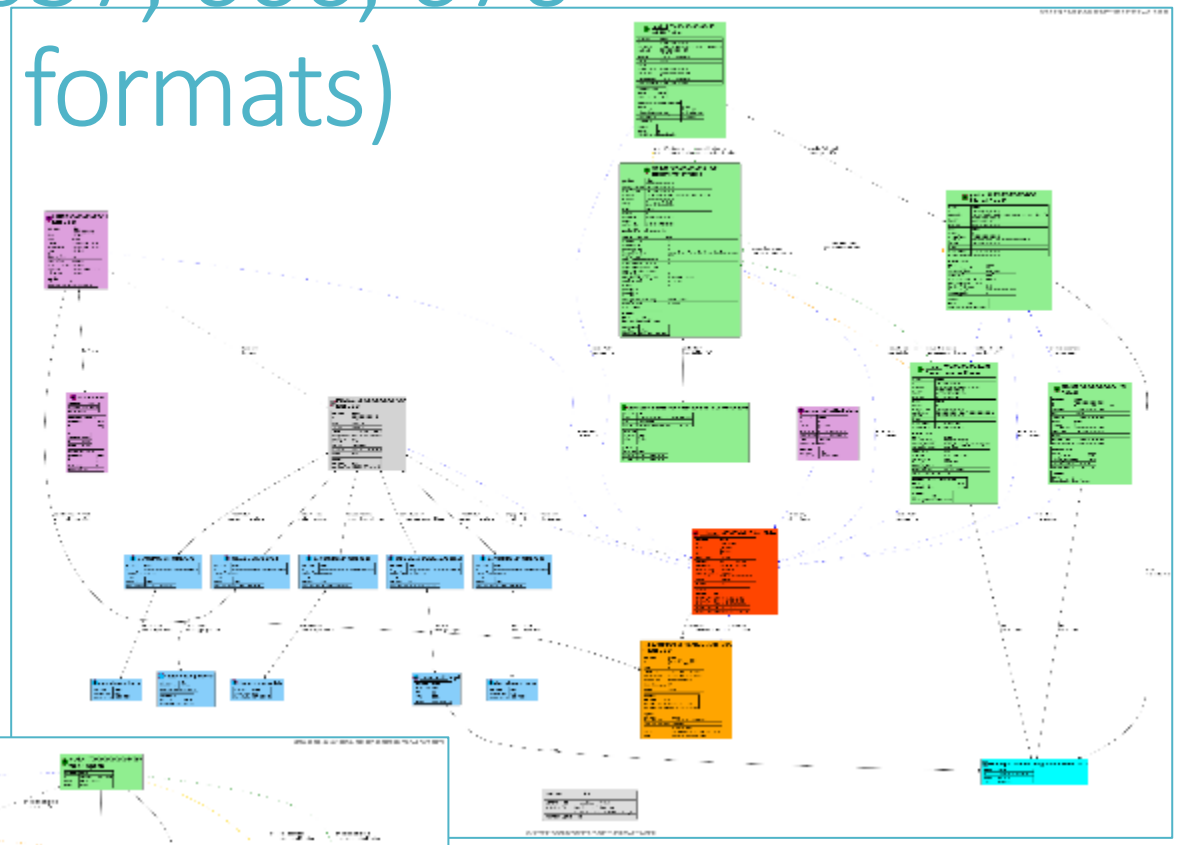


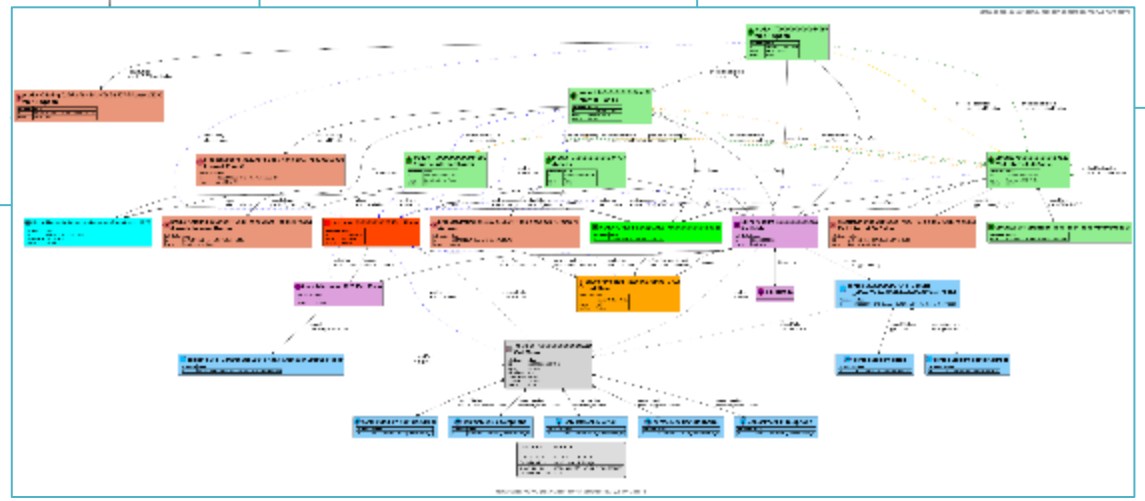
Example 2b: TMF 637 – All products of a customer



Example 3: TMF 629, 632, 637, 666, 670 for one customer (different formats)







Star Wars Example

SWAPI - The Star Wars API

https://swapi.dev/

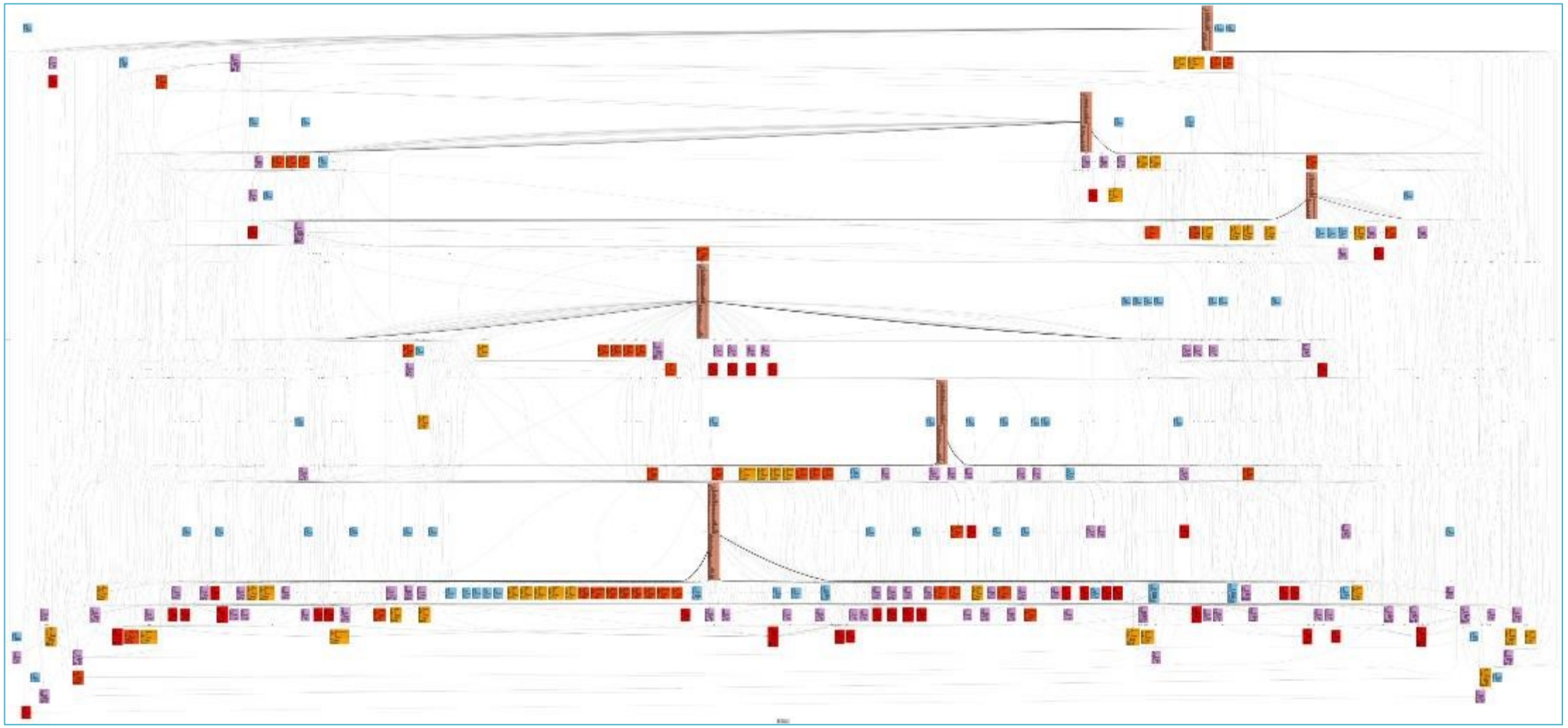
What is this?

The Star Wars API, or "swapi" (Swah-pee) is the world's first quantified and programmatically-accessible data source for all the data from the Star Wars canon universe!

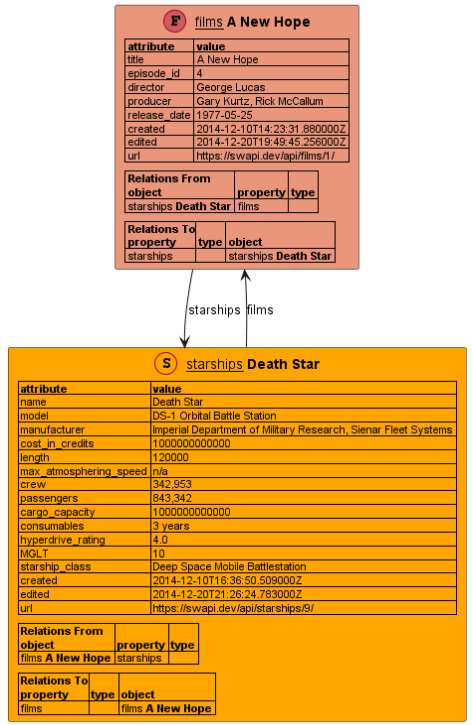
We've taken all the rich contextual stuff from the universe and formatted into something easier to consume with software. Then we went and stuck an API on the front so you can access it all!

All Data

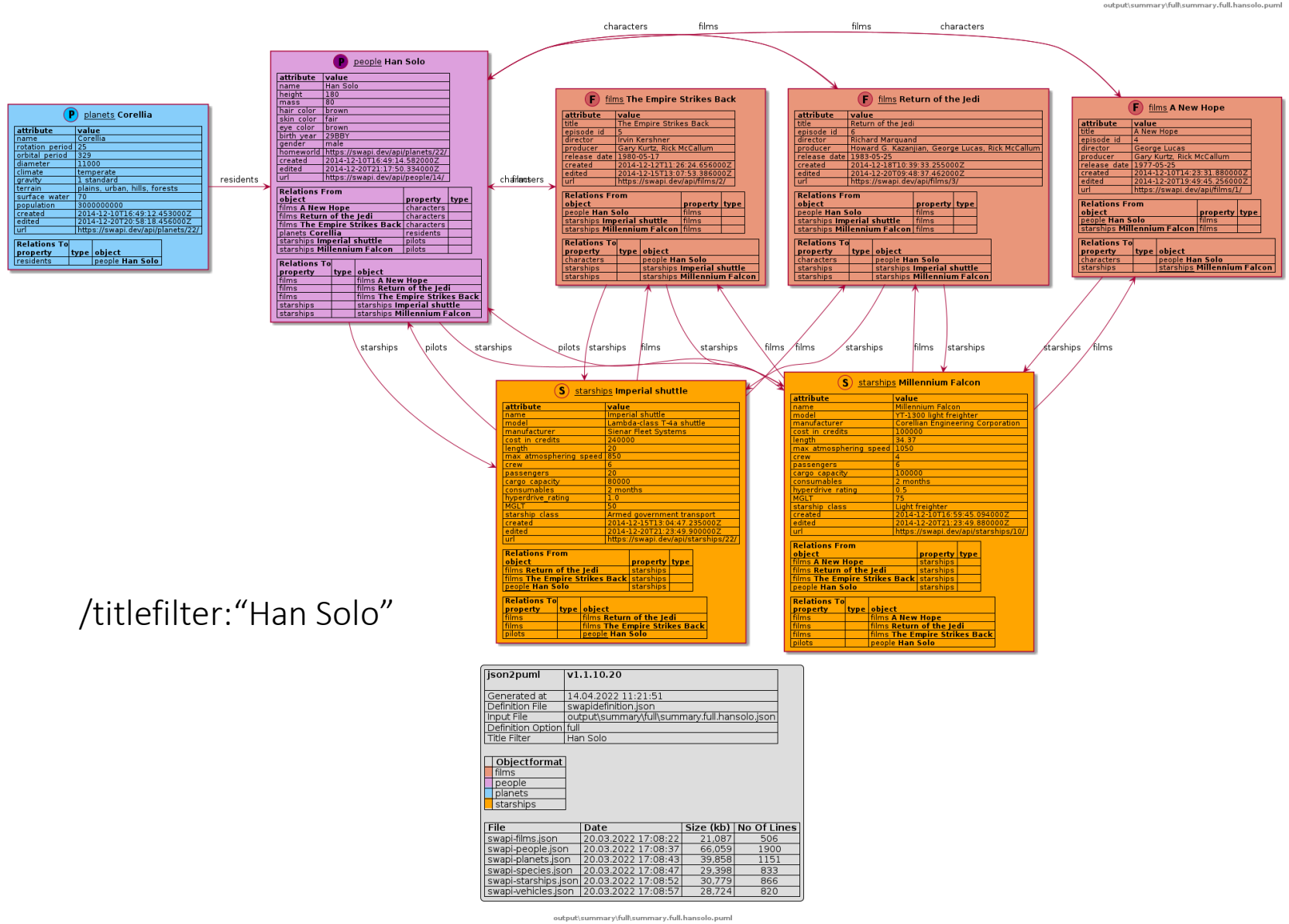
6 Films, 82 Persons, 60 Planets, 37 Species, 36 Star ships, 39 Vehicles



Examples



/titlefilter:“Death Star”



/titlefilter:“Han Solo”

C2 General

Command line parameters:

```
json2puml TMF-Open API Command Line Converter to PUML


/?                           : Help screen
/plantumljar:<file>          : Plantuml Jar file which should be used to generate the sample images
                               If defined this parameter overwrites the corresponding parameter in the definition file

/definition:<file>           : Definitionfile which contains the configuration of the mapper
/optionfile:<file>           : Optionfile which contains only the configuration of one option which then will be used for generation
/option:<name>               : Name of the option group of the definitionfile which should be used to generate the files
/alloptions                  : Flag to generate for all defined options

/inputfile:<file>            : Single JSON file to be migrated
/inputlistfile:<file>        : Listfile which contains the coniguration to handle list of
                               different files to be migrated as one big file
/leadingobject:<name>        : Name of the property which should be used as highest level of the json objects
                               This parameter is only needed for the single file conversion

/outputformat:<format>       : Format of the generated Puml converters (Allowed values: png,svg,pdf)
/openoutput:[<format>]       : Flag to define if the generated files should be opened after the generation.
                               The files will be opened using the default program to handle the file format.
                               Optional the files to be opended can be restricted by the format types (Allowed values: png,svg,pdf,puml)

/generatedetails:<boolean> : This allows to overwrite the generateDetails property of the inputlistfile
/generatesummary:<boolean> : This allows to overwrite the generateDetails property of the inputlistfile

/generateoutputdefinition  : Flag to define if the merged generator definition should be stored in the output folder.
```

Call Examples

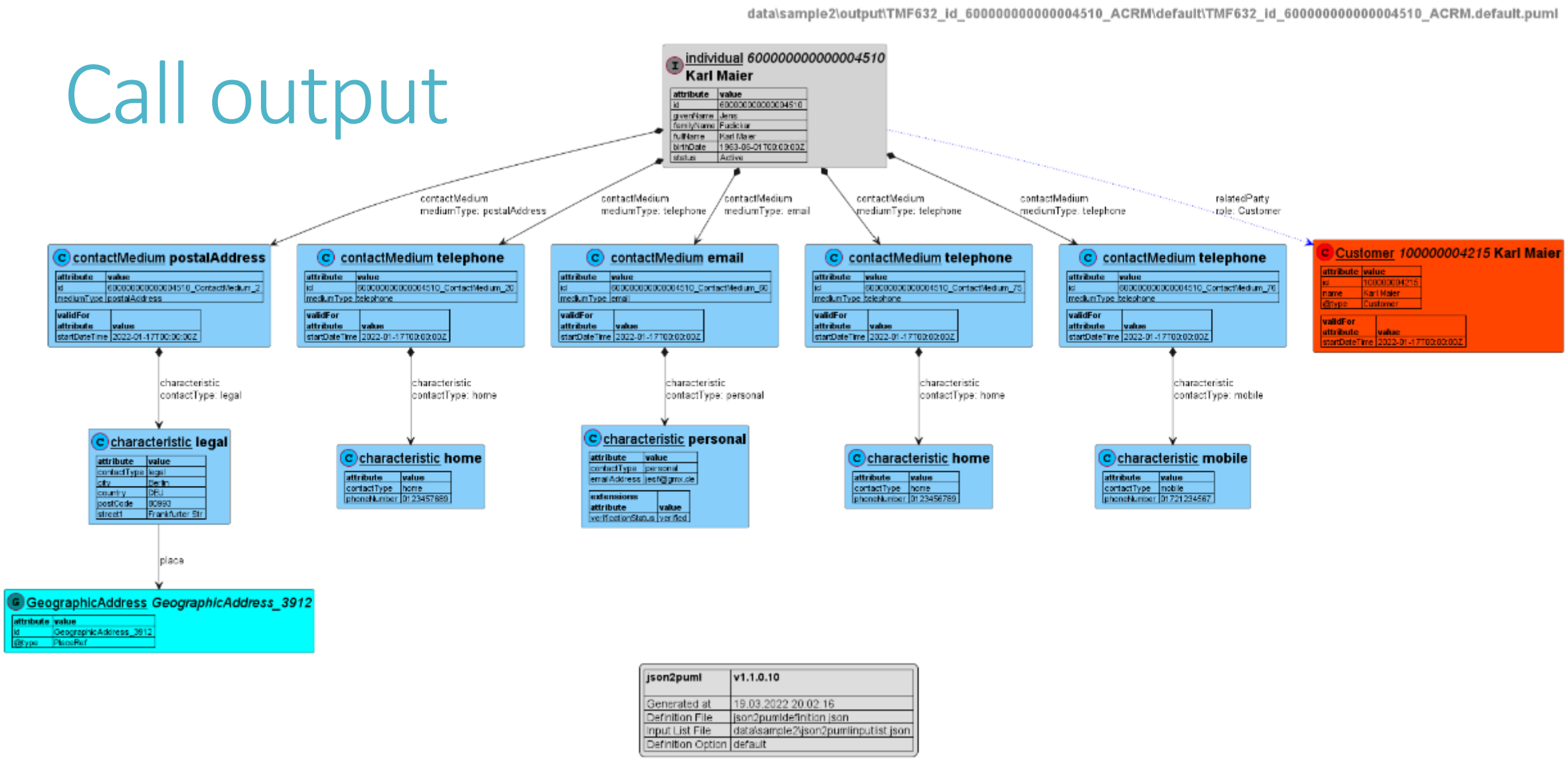
```
c:\json2puml>json2puml.exe
/inputfile:data\sample2\TMF632_id_600000000000004510_ACRM.json /leadingobject:individual
```

```
json2puml v1.0.6.6 - Command line converter JSON to PUML

Current parameters:
    /definitionfile:      definition\json2pumldefinition.json
    /inputfile:           data\sample2\TMF632_id_600000000000004510_ACRM.json
    /leadingobject:       individual

[  1/  1] Convert data\sample2\output\default\summary.default.json
               to data\sample2\output\default\summary.default.puml
               puml generated
               png  generated
               svg  generated
```

Call output


data\sample2\output\TMF632_id_600000000000004510_ACRM\default\TMF632_id_600000000000004510_ACRM.default.puml
individual 600000000000004510
Karl Maier
contactMedium
mediumType: postalAddress
contactMedium
mediumType: telephone
contactMedium
mediumType: email
contactMedium
mediumType: telephone
contactMedium
mediumType: telephone
relatedParty
role: Customer
contactMedium postalAddress
contactMedium telephone
contactMedium email
contactMedium telephone
contactMedium telephone
Customer 100000004215 Karl Maier
characteristic
contactType: legal
characteristic
contactType: home
characteristic
contactType: personal
characteristic
contactType: home
characteristic
contactType: mobile
characteristic legal
characteristic home
characteristic personal
characteristic home
characteristic mobile
place
GeographicAddress GeographicAddress_3912
json2puml
v1.1.0.10
Generated at
19.03.2022 20:02:16
Definition File
json2pumldefinition.json
Input List File
data\sample2\json2pumlinputlist.json
Definition Option
default
data\sample2\output\TMF632_id_600000000000004510_ACRM\default\TMF632_id_600000000000004510_ACRM.default.puml

Where can you find it:

Vodafone MS-Teams : json2puml
(currently restricted to people having a Vodafone Account)

Microsoft Teams - json2puml

Solstice Confluence :

Data Support Tool - json2puml - SOLSTICE - Vodafone DE Confluence