intercax

Presentation to OMG Technical Meeting, Reston, VA
Mar 20-24, 2017

SysML 2 API

Manas Bajaj - manas@intercax.com

Contributors

Ed Seidewitz - ed-s@modeldriven.com
Sanford Friedenthal - safriedenthal@gmail.com
Jeff Vodov - jvodov@draper.com
Axel Reichwein - axel.reichwein@koneksys.com

Contents

- What is an API?
- Why does SysML 2 need an API?
- What would a SysML 2 API look like?
- Can I see a Hello World example?

What is an API?

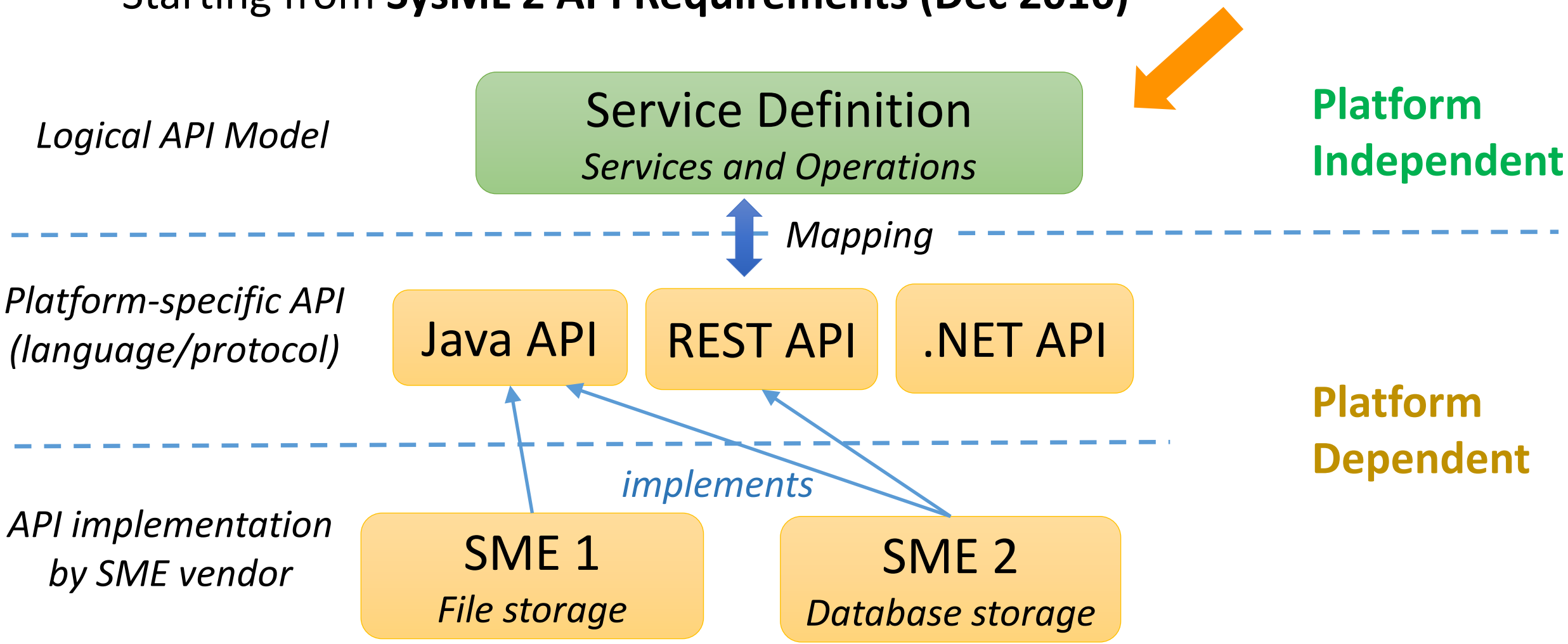
- API = Application Programming Interface
- API = Interface for software to communicate with each other
- For any software, we will typically have
  - Graphical Interface = Interface for human users
  - Application Programming Interface = Interface for other software
- What is an API useful for?
  - If a software (X) provides an API, you can write your own software (Y) to communicate with that software (X)

Why does SysML 2 need an API?

- We have always needed to access system model and automate
  - Document generation
  - Model validation
  - Model generation/transformations
  - Analysis and reasoning
- Today
  - We write plugins / scripts for a specific SysML tool
  - Application logic subject to tool-specific implementation of SysML
- SysML 2 API will make it possible to
  - Write core application (business logic) using SysML 2 services, independent of a specific tool
  - Deploy the core application for each SME
  - No need to rewrite the whole application for each SME

What would SysML 2 API look like?

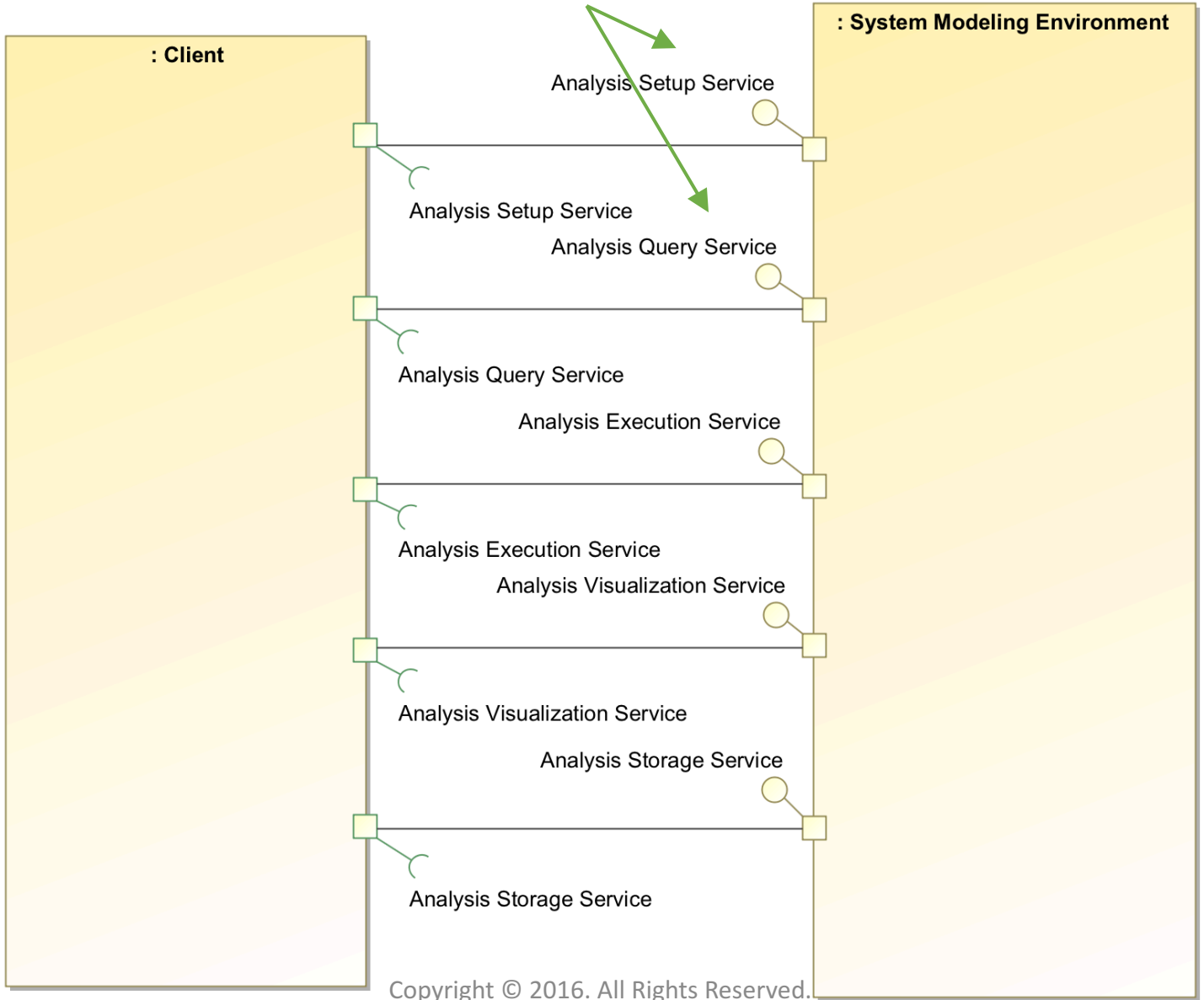
- Starting from **SysML 2 API Requirements (Dec 2016)**



SysML 2 Services – Client & Provider (SME)

SysML 2 Services

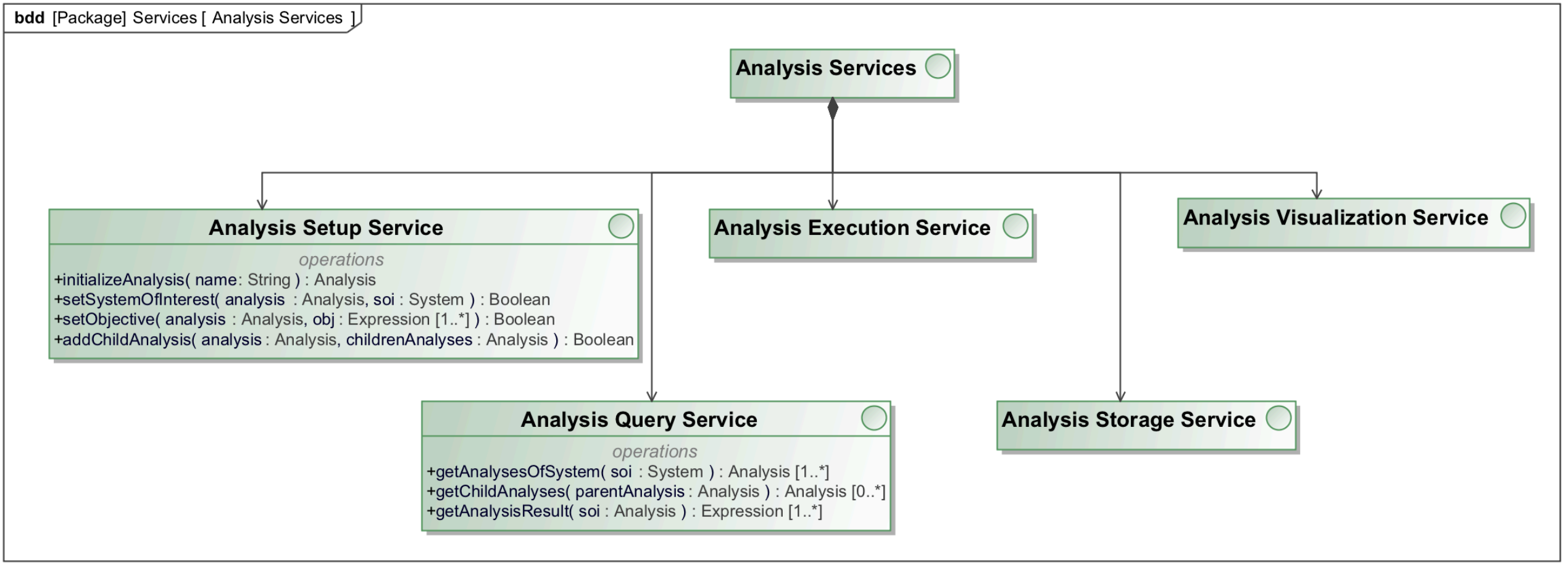
User of SysML 2 Services



SME = Provider of SysML 2 Services

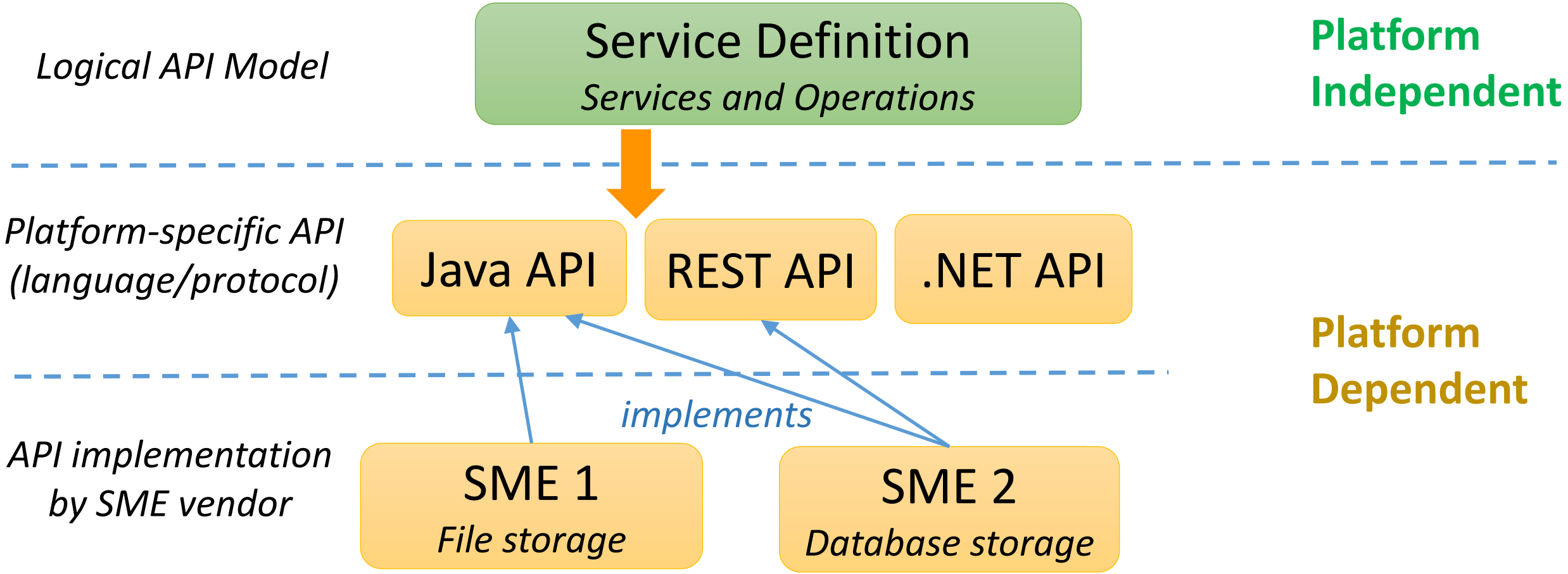
intercax

SysML 2 Service Definitions
Analysis Services Example



What would SysML 2 API look like?

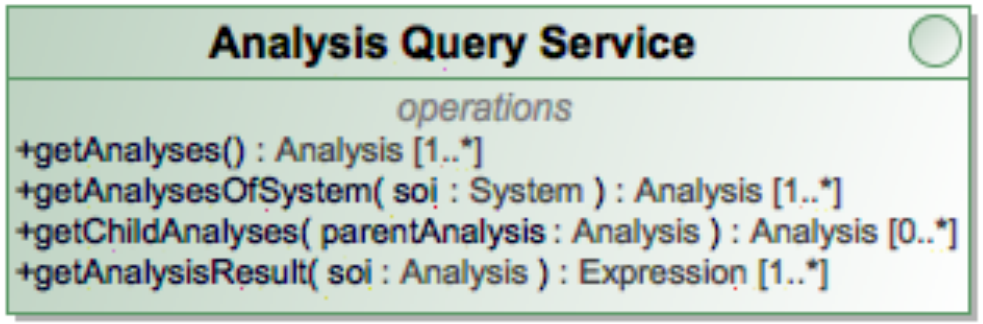
- Starting from **SysML 2 API Requirements (Dec 2016)**



Java API for Services
Java interface for each service

Service Definition

Services and Operations



Java API

```
package services;

import java.util.Collection;

import domain.Analysis;
import domain.Expression;
import domain.System;

public interface AnalysisQueryService {

    public Collection<Analysis> getAnalyses();

    public Collection<Analysis> getAnalysesOfSystem(System soi);

    public Collection<Analysis> getChildAnalyses(Analysis analysis);

    public Collection<Expression> getAnalysisResult(Analysis analysis);

}
```

intercax

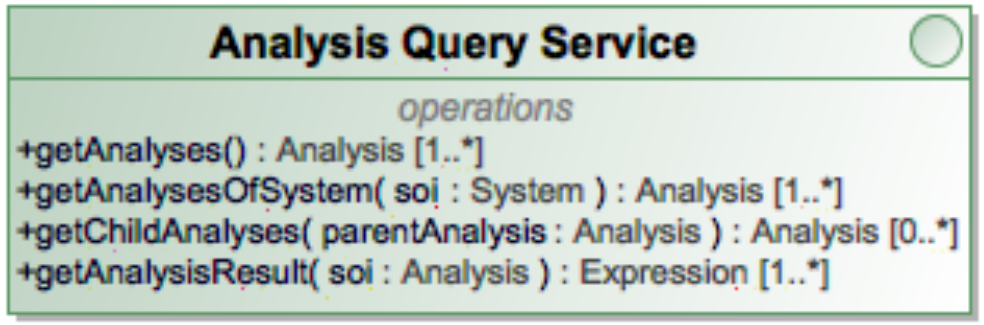
REST API for Services

Java interface for each service

Service Definition

Services and Operations

REST API



GET /sysml2/api/analyses
GET /sysml2/api/analyses/:soi
GET /sysml2/api/analyses/children/:id
GET /sysml2/api/analyses/result/:id

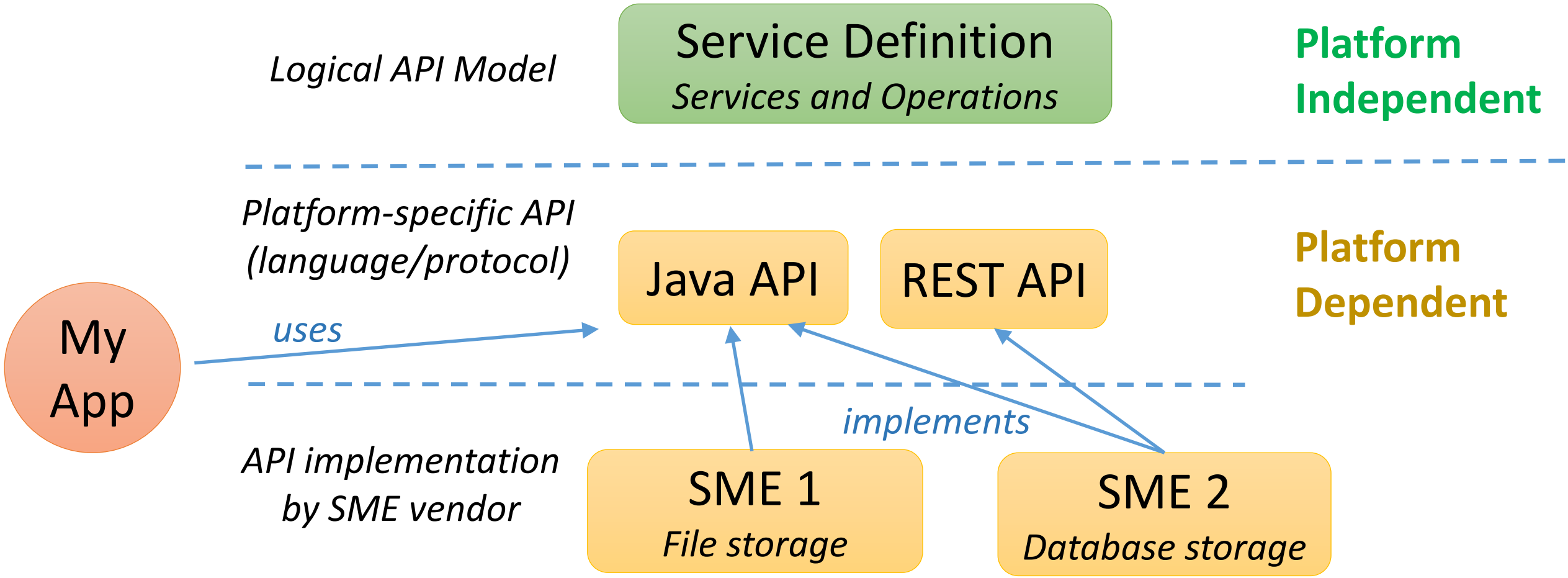
Can I see some Hello World examples?

- Example 1 – SME-independent application using SysML 2 API
  - SysML 2 Analysis Service as Java API
  - Two SMEs implement this Java API
  - Write a *simple app* using SysML 2 API and run with both SMEs

- Example 2 – REST API
  - SysML 2 Analysis Service as REST API

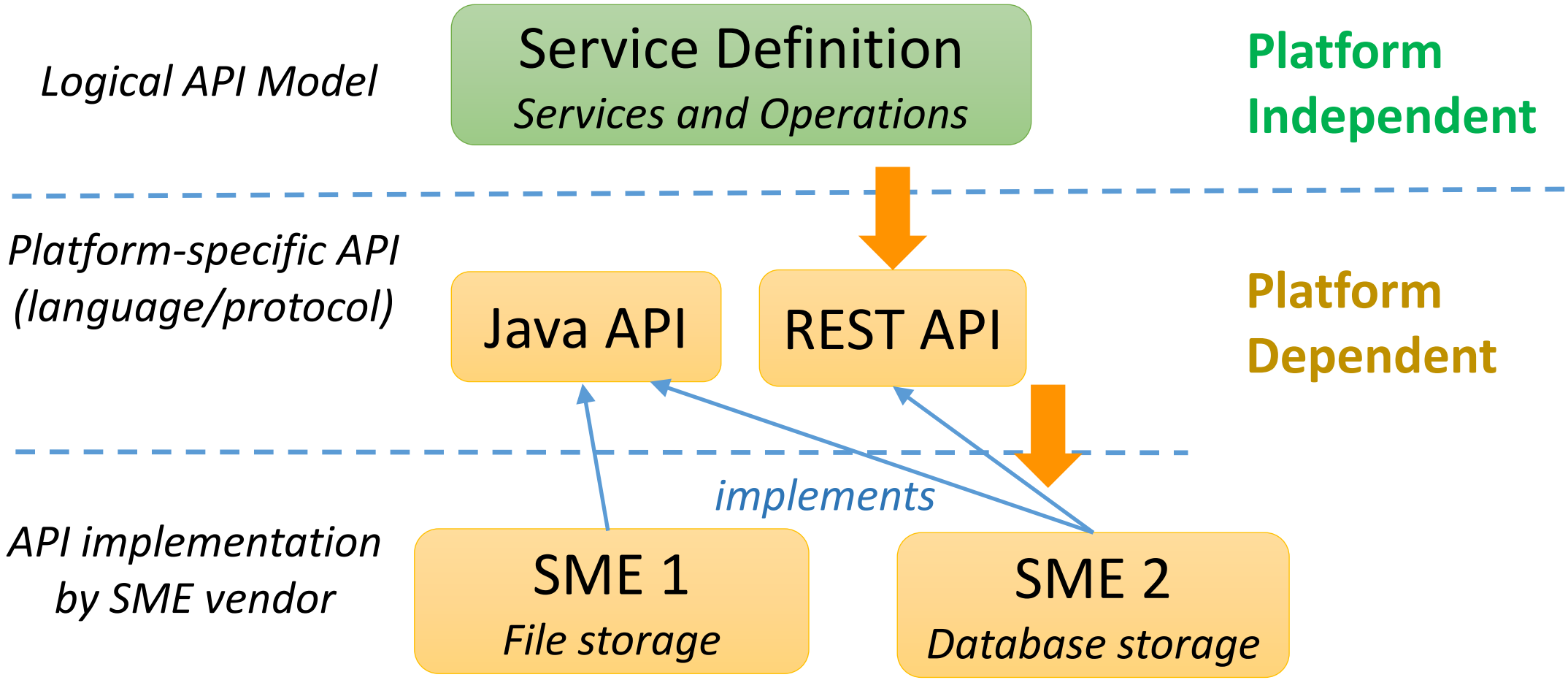
Example 1 – MyApp using SysML 2 Java API

- Starting from **SysML 2 API Requirements (Dec 2016)**



Example 2 – SysML 2 REST API

- Starting from **SysML 2 API Requirements (Dec 2016)**



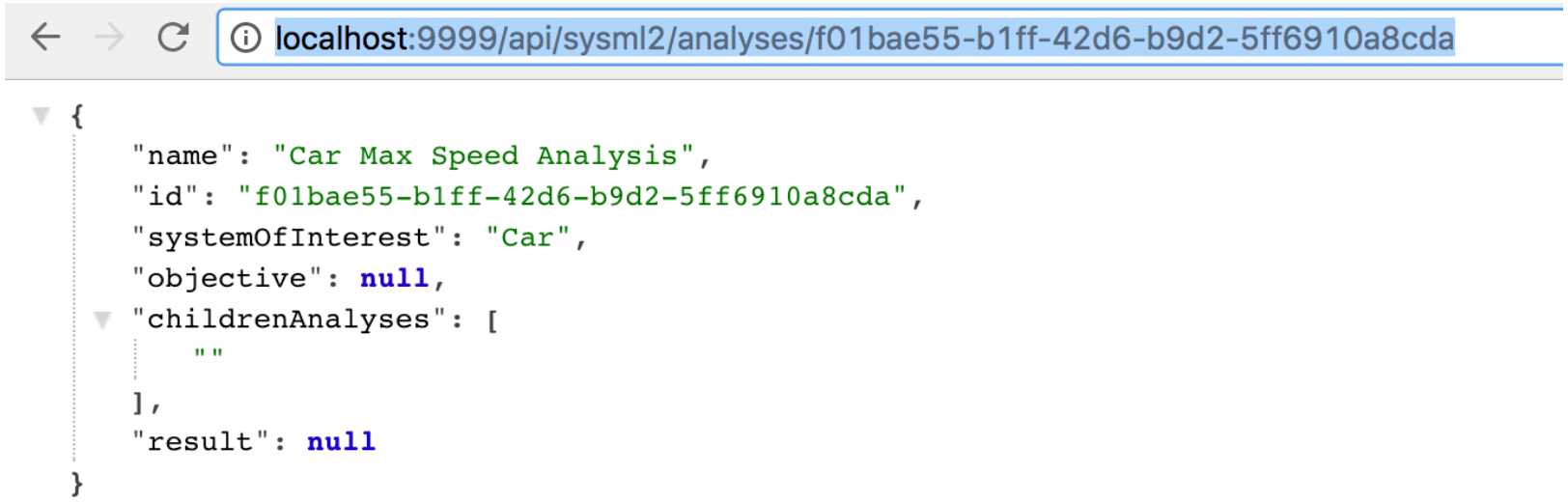
SysML 2 REST API

- Get all analyses
- **GET** <host>/api/sysml2/analyses

SysML 2 REST API (cont.)

- Get analysis by id

- **GET** <host>/api/sysml2/analyses/:**id**



SysML 2 REST API (cont.)

- Get analysis by system

- **GET** <host>/api/sysml2/analyses/system/:**id**

Questions / Comments

Manas Bajaj, PhD
Chief Systems Officer
Intercax

Email – manas.bajaj@intercax.com
Web – www.intercax.com
Voice - +1-404-592-6897, x101
LinkedIn - www.linkedin.com/in/manasbajaj
Twitter - @intercax @syndeia @manasbajaj